REPUBLIQUE DU CAMEROUN
PAIX - TRAVAIL – PATRIE
****
UNIVERSITE DE DSCHANG
****
ÉCOLE DOCTORALE

REPUBLIC OF CAMEROON
PEACE - WORK - FATHERLAND
****
UNIVERSITY OF DSCHANG
****
POST GRADUATE SCHOOL

DSCHANG SCHOOL OF SCIENCES AND TECHNOLOGY

RESEARCH UNIT IN FUNDAMENTAL COMPUTING, ENGINEERING AND APPLICATIONS (RUFCEA/URIFIA)

TOPIC:

Machine Learning for Cyber Deception in Internet of Things

Thesis presented in fulfillment of the requirements for the degree of Doctorate/PhD in Computer Science

Specialty: Computer Science

Option: Network and Distributed Services

By

MFOGO Volviane Saphir

Registration number: **CM-UDS-14SCI2035**

Master in Mathematical Sciences

Under the supervision of:

Pr NKENLIFACK Marcellin
Full Professor
University of Dschang

Pr ZEMKOHO Alain
Full Professor
University of Southampton

Dr NJILLA YAMEN Laurent
Senior Research Engineer
Air Force Research Laboratory

YEAR 2024

Certification of the originality of the work

I the undersigned MFOGO Volviane Saphir, student in the Department of Mathematics and Computer Science, Faculty of Sciences, University of Dschang, certify that the present work titled **Machine Learning for Cyber Deception in Internet of Things**, carried out in the Research Unit of the Fundamental Computer Science, Engineering and Applications (URIFIA) of the University of Dschang, under the supervision of Pr. NKENLIFACK Marcellin, Pr. ZEMKOHO Alain and Dr. NJILLA YAMEN Laurent is original.

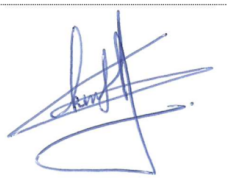
This has never previously been published for the award of any degree.

The Student

MFOGO Volviane Saphir

Date: ……..

The Directors

		
Pr. Marcellin Nkenlifack Associate Professor University of Dschang Cameroon *Date: Sep 23, 2024*	**Pr. Alain Zemkoho** Full Professor at School of Mathematical Sciences University of Southampton United Kingdom *Date: Sep 25, 2024*	**Dr. Laurent Njilla** Senior Research Engineer Air Force Research Laboratory USA *Date: Sep 23, 2024*

The Head of Department

Date:

Dedication

To my parents: my father KENFACK Volvian De L'or
and my mother KONGNEK Veronique.

Acknowledgments

This work would not have been possible without life, good health, courage, inspiration, and grace from the Almighty God.

Academic Acknowledgments

- First of all, I would like to express my sincere and deepest gratitude to my supervisors, **Professor NKENLIFACK Marcellin, Professor ZEMKOHO Alain, and Dr. NJILLA YAMEN Laurent**. During these three years, they supervised my thesis in an exceptional and remarkable way. I especially thank them for suggesting this interesting topic, for many helpful conversations every week, for their comments that improved the final output, and for reading and correcting drafts of this work many times.

- This research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-21-1-0326. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

- I extend my warm appreciation and gratitude to **DEVCOM ARL Army Research Office** for sponsoring this research and to all members of the **GMC-DRA Project**, especially **Dr. KAMHOUA Charles** from DEVCOM Army Research Laboratory, for the good atmosphere, advice, support, and understanding he showed me throughout these three years.

- My gratitude also goes to all the lecturers of the **Department of Mathematics and Computer Science, Faculty of Science, University of Dschang**. Special thanks to **Professor KENGNE TCHENDJI Vianney** for reviewing this work.

- To the jury members and experts, I sincerely thank you for accepting to participate in the evaluation of this work.
- I especially want to thank my academic elders for their support and encouragement, particularly **Dr. FOKO Miguel Landry**.

Family Acknowledgments

- My greatest acknowledgment goes to my beloved parents, **Mr. KENFACK Volvian De L'or and Mrs. KONGNEK Véronique**. Your unwavering love, patience, and faith have been my greatest source of strength. Your sacrifices, encouragement, and belief in my dreams have shaped the person I am today. This achievement is as much yours as it is mine.
- My father-in-law, **Mr. NAOUSSI Michel**, thank you for trusting me and for your kind words. I also thank my mother-in-law, **Mrs. DJOUALA Marcelline**, for her great support.
- To my brothers and sisters: **Mr. TANKEU Jaures Cyrille, Mr. LIENOU KENFACK Volvian Dumand, Mr. WAMBA FOGOUG Volvian Silvansien, Miss DJUIPET KENFACK Elvira, Mr. CHEUDHEU Volvian Kerlin, and Miss MALIEMBA Marie Josseline D'or**—your support has meant the world to me.
- Special thanks to my sister **Miss MEDONGANG KONGNEK Volviane Laura** for being there during the hardest times.
- To my grandmother, **Mamy MEDONGANG Thérèse**, and my aunts, **Mamy MBOGNING Régine and Mrs. MANEKENG Émelie Laure**, thank you for your prayers and unwavering support.
- To my brothers- and sisters-in-law, thank you for your motivation and encouragement. Special thanks to **Mrs. MAFFO Blandine** for her unconditional support, especially when I was preparing for conferences.

Friends and Community

- Thanks to my dear friends: **Dr. FOUTSE YUEHGOH, Dr. WAMBA Lyse Noamie, Dr. MOSKOLAÏ Justin, and Mr. NANDO TEZOH Franky** for your inspiring presence and steadfast support.
- To the **Deep Learning Indaba** community and the **IndabaX Cameroon** community—thank you for your encouragement and collaboration. Special thanks to **Mrs. Amal Rannen-Triki** for being a role model in community building, for your guidance, and for your unwavering support.
- Thanks to all my lab mates for the team spirit and the life we shared together.

Personal Acknowledgments

- I would like to give great and special appreciation to my beloved husband, **Mr. ATENEBEUN Stanley Wilson**, for his infinite listening, moral support, and unwavering encouragement. Your belief in me has been my greatest motivation.
- Last but certainly not least, my dearest and most precious daughters, **ATENEBEUN DJOUMESSI Lylle Johanna and ATENEBEUN MFOGO Evy Lena**. Your arrival during my PhD journey was the greatest blessing of all. Your love, laughter, and presence gave me the strength to push forward even in the most challenging moments. This achievement is also for you, my little angels—may it inspire you to chase your dreams with the same determination and passion.

I thank all those who helped me, near or far, to realize this work. May God bless you all.

Table of Contents

List of Symbols xi
List of Acronyms xii
List of Tables xiv
List of Figures xv
Abstract xvii
Résumé xix

General Introduction **1**
Context of the Work 1
Problem Statement 2
Research Questions 2
Objectives 3
General Objective 3
Specific Objectives 3
Key Contributions 4
Structure of the Dissertation 5

Chapter I ► **Background on Internet of Things, Cyber Deception, and Machine Learning** **8**
I.1 - Introduction 8
I.2 - Internet of Things 9
I.2.1 - Overview of IoT 9
I.2.2 - IoT Architectures 10
I.2.3 - IoT Technologies and Protocols 12
I.2.4 - IoT Networks Compared to Wireless Sensor Networks and Cloud 13
I.2.5 - IoT Applications and Use Cases 14
I.2.6 - IoT Security Challenges 14
I.2.7 - Traditional Network Security vs IoT Security 15
I.3 - Cyber Deception 16
I.3.1 - Definition and Concepts 17
I.3.2 - Multidisciplinary Perspectives on Deception 17

I.3.3 - Motivations for Cyber Deception 18
I.3.4 - Types and Properties of Cyber Deception 18
I.3.5 - Distinctions between Defensive Deception and Other Defense Techniques 19
I.4 - Machine Learning 20
I.4.1 - Overview of Machine Learning 20
I.4.2 - Types of Learning 21
I.4.3 - Deep Learning 22
I.4.4 - Applications of ML in Cybersecurity 26
I.5 - Conclusion 26

Chapter II ► **Related Works on Defense Strategies for IoT Security using Machine Learning** **28**
II.1 - Introduction 28
II.2 - Overview of IoT Security 29
II.2.1 - Identification of Security and Privacy Issues 30
II.2.2 - Biometric Authentication 31
II.2.3 - Blockchain for IoT Security 32
II.2.4 - Layered Security Approach 32
II.3 - Security Threats on IoT Devices 33
II.3.1 - Reconnaissance Attacks 35
II.3.2 - Vulnerability Exploitation 35
II.3.3 - Data Exfiltration 36
II.4 - Machine Learning for IoT Security 37
II.4.1 - Anomaly Detection 38
II.4.2 - Intrusion Detection 38
II.4.3 - Predictive Analytics and Malware Detection 39
II.4.4 - Cloud-Assisted Security Services 39
II.5 - Cyber Deception for IoT Security 40
II.5.1 - IoT Honeypots in Cybersecurity 41
II.5.1.1 - Low-Interaction Honeypots 42
II.5.1.2 - High-Interaction Honeypots 42
II.5.1.3 - Medium-Interaction Honeypots 43
II.5.1.4 - Intelligent-Interaction Honeypots 43
II.5.2 - Fake Vulnerability for IoT Device 45
II.5.3 - Honeypot for IoT Camera 45
II.6 - Conclusion 46

Chapter III ► **Contribution to Adaptive Intelligent Cyber Deception for Probing Attacks on IoT Devices** **48**
III.1 - Introduction 48
III.2 - Probing Attacks on IoT Devices 49
III.2.1 - Phases of Probing Attacks 50
III.2.1.1 - Reconnaissance Phase 50
III.2.1.2 - Scanning Phase 51
III.2.1.3 - Enumeration Phase 51
III.2.2 - Techniques Used in Probing Attacks 51
III.2.2.1 - IPsweep 51
III.2.2.2 - Nmap Scans 52
III.2.2.3 - Portsweep 52
III.2.3 - Impact of Probing Attacks on IoT Security 52
III.2.3.1 - Exploitation of Vulnerabilities 52
III.2.3.2 - Preparation for Sophisticated Attacks 53
III.2.3.3 - IoT Device Vulnerabilities 53
III.2.3.4 - Pre-check Attack Step 53
III.3 - Adaptive Intelligent Interaction Honeypot for Internet of Things . . 54
III.3.1 - System Overview 55
III.3.2 - Req/Res Database and Request Evaluator 56
III.3.3 - Intelligent Interaction Honeypot: Honey-Chatbot 57
III.3.3.1 - BERT-based Honeypot 57
III.3.3.2 - GAN & BERT-based Honeypot 58
III.3.4 - Adaptive Intelligent Interaction Honeypot 61
III.3.4.1 - Model Formulation 61
III.3.4.2 - Adaptive Mechanisms 61
III.3.5 - Request evaluator 64
III.4 - Evaluation and Results 65
III.4.1 - Evaluation of the Learning Model 65
III.4.2 - Result of the Effectiveness of GAN-AIIPot 66
III.5 - Analysis and Discussions 69
III.5.1 - Analysis 70
III.5.2 - Discussions 71
III.6 - Conclusion 73

Chapter IV ► **Contribution to Adaptive Learning for IoT Security: A Hybrid Recommender System Approach** **74**

IV.1 - Introduction . . . 74
IV.2 - Preliminaries . . . 75
IV.2.1 - Vulnerability on IoT Devices . . . 75
IV.2.2 - Recommender Systems . . . 76
IV.3 - Adaptive Learning-Based Hybrid Recommender Deception System . 77
IV.3.1 - System Design and Architecture . . . 77
IV.3.2 - Algorithms Used . . . 77
IV.3.3 - Overview . . . 79
IV.3.4 - Detailed Structure of VARIoT Database . . . 81
IV.3.4.1 - IoT Vulnerabilities . . . 81
IV.3.4.2 - IoT Exploits . . . 82
IV.3.5 - Adaptive Learning-Based Hybrid Recommender System (ALHRS) . . . 82
IV.3.5.1 - Collaborative and Content-Based Filtering Mechanisms 82
IV.3.5.2 - Adaptive Learning for Weight Optimization . . . 84
IV.3.6 - Deception Framework . . . 85
IV.3.7 - Use Case and Implementation . . . 88
IV.3.7.1 - System Integration and Workflow: Threat Model . . 88
IV.3.7.2 - Example Scenario: Securing a Surveillance Camera 88
IV.4 - Evaluation and Results . . . 89
IV.4.1 - Accuracy Metrics . . . 89
IV.4.2 - Effectiveness of Deception . . . 90
IV.4.3 - Scalability and Response Efficiency . . . 93
IV.5 - Analysis and Discussions . . . 93
IV.5.1 - Analysis . . . 94
IV.5.2 - Discussions . . . 95
IV.6 - Conclusion . . . 96

Chapter V ► **Contribution to Advanced Security using Augmented Reality Honeypots: A Case Study for IoT Cameras** **98**
V.1 - Introduction . . . 98
V.2 - IoT Cameras and Augmented reality . . . 99
V.2.1 - IoT Cameras . . . 100
V.2.2 - Augmented Reality . . . 100
V.3 - Adaptive Honeypot-based Augmented Reality for IoT Camera: ARPotCam . . . 101
V.3.1 - System Overview . . . 102

V.3.1.1 - Frontend Honeypot IoT Camera 103
V.3.1.2 - Backend Real IoT Camera 104
V.3.1.3 - Dynamic Behavior Emulation Through RL 105
V.3.1.4 - AR Processing Unit 109
V.3.2 - System Implementation 113
V.4 - Evaluation and Results 114
V.4.1 - System Implementation 114
V.4.1.1 - Configuring Video Streams 114
V.4.1.2 - Creation of AR Environment 114
V.4.1.3 - Communication and Interaction 116
V.4.2 - Experimental Result 116
V.5 - Analysis and Discussions 124
V.5.1 - Analysis 124
V.5.2 - Discussions 126
V.6 - Conclusion 127

General Conclusion **128**
Problem Studies 128
Summary of Research Findings 128
Limitations of the Study 130
Future Research Directions 131

Bibliography **133**

Appendix A ► **Setting Up a Honeypot on Google Cloud Platform** **a**

Appendix B ► **List of Publications** **e**

List of Symbols

$\mathbb{X}$	A set of input dataset;
Y	A set of label/output dataset;
a	Action;
s	Current state;
s'	Next state;
t	Timestamp;
R	Reward;
x_t, y_t, w_t	Variables representing state, response, and weights in a model at time t;
$\pi(y_t \mid x_t)$	Probability of choosing response y_t given the input request x_t;
$\theta_{CF}, \theta_{CBF}$	Adaptive weights assigned to the outputs from Collaborative Filtering and Content-Based Filtering.

List of Acronyms

AI	Artificial Intelligence;
AL-HRS	Adaptive Learning-based Hybrid Recommender System;
AR	Augmented Reality;
ARPotCam	Augmented Reality honeyPot Camera;
BERT	Bidirectional Encoder Representations from Transformers;
BLE	Bluetooth Low Energy;
CBF	Content-Based Filtering;
CF	Collaborative Filtering;
CoAP	Constrained Application Protocol;
CVE	Common Vulnerability Exposure;
CVSS	Common Vulnerability Scoring System;
DL	Deep Learning;
DLN	Deep Learning Network;
DDoS	Distributed Denial of Service;
DQN	Deep Q-Network;
DoS	Denial of Service;
GAN	Generative Adversarial Networks;
GCP	Google Cloud Platform;
IP	Internet Protocol;
IoT	Internet of Things;
IT	Information Technology;
LSTM	Long Short-Term Memory;
ML	Machine Learning;
MDP	Markov Decision Process;
M2M	Machine to Machine;
MTD	Moving Target Defense;
MQTT	Message Queuing Telemetry Transport;
Nmap	Network Mapper;
NSL-KDD	NSL-Knowledge Discovery and Data mining Dataset;
OOV	Out-of-Vocabulary;

PBR	Policy-Based Routing;
R2L	Remote to Local;
Req/Res	Request/Response;
RL	Reinforcement Learning;
RTMP	Real-Time Messaging Protocol;
RTMPS	Secure Real-Time Messaging Protocol;
SIPHON	High-Interaction IoT Honeypot Architecture;
TCP	Transmission Control Protocol;
TTPs	Tactics, Techniques, and Procedures;
U2R	User to Root;
UDP	User Datagram Protocol;
WSN	Wireless Sensor Network.

List of Tables

Table I - Applications of ML in Cybersecurity 37
Table II - ML Techniques for IoT Security & Honeypots 40

Table III - Times of experimentation for each model. 67

Table 4 - Comparison of accuracy metrics across different systems 90
Table 5 - Recommended vulnerabilities for smart home devices. 90
Table 6 - Evaluation of the attractiveness of the deception system. 93

Table 7 - Number of requests received by each honeypot on the GCP web server 116
Table 8 - Types of attack interactions captured 117
Table 9 - Port scan distribution 118
Table 10 - Scalability performance metrics 123
Table 11 - Delay measurements for augmented video stream 124

List of Figures

Figure 1 - The evolution of the internet from pre-internet to IoT [1]. 9
Figure 2 - Five-layer architecture of IoT [2] 12
Figure 3 - A Venn diagram depicting the nested relationship between Cyber Security, Cyber Deception, Defensive Deception, and Honey-X. Cyber Security is the broadest category, within which Cyber Deception is a subset, followed by Defensive Deception, with Honey-X being a specific implementation within Defensive Deception [3]. 16
Figure 4 - A Venn diagram illustrating the hierarchical relationship between Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL). AI encompasses the broadest field, within which ML is a subset, and DL is a further subset within ML [4]. 20
Figure 5 - IoT attack classification [1]. 34
Figure 6 - Life cycle of IoT attack. 50
Figure 7 - High-level overview of AIIPot. 55
Figure 8 - Req/res database entries. 56
Figure 9 - Request/response multi-entries. 56
Figure 10 - BERT with GAN architecture: G generates a set of fake responses F given a random distribution of requests. These, along with the real req/res dataset vector representations computed by BERT, are used as input for the discriminator D. 60
Figure 11 - Learning curve of the model with different fusion methods. . . 65
Figure 12 - Learning curve of the proposed GAN & BERT model with the best fusion method. 66
Figure 13 - GAN & BERT model response on unseen request data. 66
Figure 14 - Session length comparison between attackers and honeypots (in percentage). 68
Figure 15 - The amount of information received by the proposed honeypot per session length. 69

Figure 16 - The count of attacks detected by the honeypot categorized according to the NSL-KDD dataset. 70

Figure 17 - System architecture overview. 80
Figure 18 - VARIoT database entries. 81
Figure 19 - Comparison of attacker engagement times with recommended vulnerabilities by AL-HRS and existing method. 91
Figure 20 - Correlation between engagement time and volume of information sent by attackers, indicating the effectiveness of the recommended vulnerabilities in extracting valuable intelligence. 92
Figure 21 - Detection rate of AL-HRS showing the system's effectiveness in identifying and responding to well-known attack types such as denial of service (DoS), probe, user to root (U2R), and remote to Local (R2L). 92

Figure 22 - General overview of the model. 102
Figure 23 - AR processing unit steps. 109
Figure 24 - Flowchart of the augmented reality processing unit. 111
Figure 25 - Example of AR video frame under the action "remove the car". 115
Figure 26 - Duration an attacker spends interacting with the honeypot. . . . 119
Figure 27 - Number of camera control commands issued by the attacker. . . 120
Figure 28 - Varieties of camera control command distribution. (a) Real-Camera Honeypot [5], (b) HoneyCam [6], (c) ARPotCam 121
Figure 29 - Proportion of attackers misled by the honeypot. 121
Figure 30 - Frequency and success of adaptive responses generated by the RL.122
Figure 31 - Covertness Evaluation: Percentage of Attackers Deceived . . . 124

Figure 32 - Flow diagram of the interaction between our objectives, contributions and simulations . . . . . d

Abstract

The proliferation of the Internet of Things (IoT) has interconnected a vast collection of devices, from household appliances to industrial machinery, facilitating seamless data exchange and task automation. However, this extensive connectivity introduces significant security vulnerabilities, as IoT devices often lack robust security measures due to their limited computational resources and diverse operating environments. Traditional security solutions, which are generally reactive, fail to provide comprehensive protection against sophisticated and evolving cyber threats. This thesis addresses the pressing need for innovative and proactive security measures to protect IoT ecosystems. Our research focuses on integrating machine learning (ML) and cyber deception techniques to improve the security of IoT devices. Our objective is to propose and develop a comprehensive security framework designed to detect and mislead attackers, thereby providing real-time protection against potential threats. This thesis's key contributions include the development of an adaptive intelligent interaction honeypot system, a hybrid recommender system leveraging adaptive learning, and an augmented reality-based honeypot for IoT cameras. The adaptive intelligent interaction honeypot system is designed to engage and mislead attackers during the reconnaissance phase, capturing valuable data on attack methods and tactics. This data is used to dynamically adapt the honeypot's deception capabilities, making it more challenging for attackers to distinguish between real and decoy devices. The hybrid recommender system integrates adaptive learning to propose both real and fake vulnerabilities, confusing and delaying attackers and reducing the likelihood of successful breaches. Finally, the augmented reality-based honeypot creates realistic, interactive decoys for IoT cameras, engaging attackers for extended periods and gathering detailed intelligence on their strategies and tools. Our findings demonstrate the adequateness of traditional security measures in adapting to protect IoT devices from advanced cyber threats and highlight the effectiveness of innovative approaches that combine ML and cyber deception. By addressing the gaps identified in the current literature and providing practical applications of these techniques, this research offers valuable insights and tools to improve IoT security.

Keywords: Internet of Things, Security, Machine Learning, Cyber Deception, Honeypots, Reinforcement Learning, Augmented Reality, Hybrid Recommender System.

Apprentissage Automatique pour la Cyber Déception dans l'Internet des Objets

Résumé

L'Internet des objets permet de connecter une multitude de dispositifs, allant des appareils domestiques aux machines industrielles. Il rend l'échange de données et l'automatisation des tâches plus aisés de nos jours. Toutefois, cette connexion étendue engendre des lacunes de sécurité importantes, car les appareils d'Internet des objets manquent fréquemment de mesures de sécurité solides en raison de leurs ressources computationnelles restreintes et de leurs environnements de fonctionnement variés. Les solutions de sécurité traditionnelles, généralement réactives, ne parviennent pas à offrir une protection complète contre les menaces de la cyber sécurité sophistiquées et en constante évolution. Cette thèse répond à la nécessité urgente de mesures de sécurité innovantes et proactives pour protéger les écosystèmes d'Internet des objets. Notre recherche se concentre sur l'intégration des techniques d'apprentissage automatique et de cyber déception pour améliorer la sécurité des dispositifs d'Internet des objets. Nous proposons une mesure de sécurité complète conçue pour détecter et leurrer les attaquants, offrant ainsi une protection en temps réel contre les menaces potentielles. Les contributions clés de cette thèse incluent le développement d'un système de pot de miel interactif intelligent et adaptatif, d'un système de recommandation hybride utilisant l'apprentissage adaptatif et d'un pot de miel basé sur la réalité augmentée pour les caméras intelligentes. Le système de pot de miel interactif, intelligent et adaptatif est conçu pour engager et leurrer les attaquants lors de la phase de reconnaissance du réseau, capturant des données précieuses sur les méthodes et tactiques d'attaque. Le système de recommandation hybride intègre l'apprentissage adaptatif pour proposer à la fois des vulnérabilités réelles et fictives, retardant les attaquants, et réduisant ainsi la probabilité des attaques réussies. Enfin, le pot de miel basé sur la réalité augmentée crée des leurres interactifs et réalistes pour les caméras intelligentes, interagissant avec les attaquants pendant de longues périodes et recueillant des renseignements détaillés sur leurs stratégies et outils. Nos résultats mettent en évidence le manque de capacité des mesures de sécurité classiques à protéger de manière adaptative les dispositifs d'Internet des objets contre les risques de cyber sécurité avancés, et soulignent l'efficacité de notre approche novatrice.

Mots clés: Internet des Objets, Sécurité, Apprentissage Automatique, Cyber Déception, Pot de miel, Apprentissage par Renforcement, Réalité Augmentée, Système de Recommandation Hybride.

General Introduction

CONTENTS

Context of the Work 1
Problem Statement 2
Research Questions 2
Objectives 3
Key Contributions 4
Structure of the Dissertation 5

Context of the Work

The rapid expansion of the Internet of Things (IoT) has led to a world where billions of interconnected devices facilitate automation and real-time data exchange across various domains, including healthcare, smart homes, and industrial systems. This unprecedented connectivity has revolutionized various sectors by improving efficiency, enhancing user experiences, and enabling intelligent automation. However, despite these transformative benefits, IoT systems suffer from inherent security weaknesses that pose severe threats to privacy, data integrity, and system reliability. IoT devices often operate with limited computational resources, lack robust security mechanisms, and communicate over unsecured networks, making them prime targets for cyber attackers [7]. The rapid proliferation of IoT technology, coupled with the increasing sophistication of cyber threats, underscores the urgent need for innovative security solutions capable of proactively mitigating these risks before they escalate into widespread attacks.

Manufacturers, driven by the race to innovate and commercialize IoT products quickly, often compromise security by embedding hard-coded credentials, offering minimal user-controlled security configurations, and failing to provide timely firmware updates. These weaknesses make IoT devices particularly vulnerable

to exploitation, leading to unauthorized access, data breaches, and service disruptions. Unlike conventional computing systems, where security measures such as firewalls and endpoint protection are well-established, IoT environments require adaptive and intelligent security mechanisms capable of addressing evolving threats.

Traditional security approaches are largely reactive, responding to attacks after they have occurred. This approach is inadequate in the face of sophisticated cyber threats that leverage automation and artificial intelligence to exploit vulnerabilities in real time. The growing number of security breaches in IoT ecosystems underscores the urgent need for proactive, intelligent, and deceptive security measures to mitigate risks and protect sensitive data.

Problem Statement

Despite continuous advancements in IoT security, existing solutions remain insufficient against the evolving landscape of cyber threats. Conventional security mechanisms, such as signature-based intrusion detection systems, struggle to detect novel attack patterns. The increasing scale and complexity of IoT networks further exacerbate these challenges, creating an expanding attack surface that adversaries exploit.

A critical limitation of traditional security solutions is their reactive nature, as they primarily focus on detecting and responding to attacks post-compromise. This delayed response allows attackers to achieve their objectives before countermeasures are deployed. Furthermore, many IoT security strategies lack adaptability, failing to adjust dynamically to emerging threats in real time. These deficiencies highlight the need for an innovative security framework that not only detects cyber threats proactively but also deceives and engages attackers, delaying or preventing them from causing harm.

Research Questions

Building upon the foundational work identified in [8], the following research questions guide the scope and direction of this work:

- What are the common vulnerabilities in IoT devices, and how can they be exploited by attackers?

- How can machine learning techniques be applied to detect cyber threats targeting IoT devices?
- What cyber deception techniques can be designed to effectively mislead and capture attackers?
- How effective is the machine learning security framework in protecting IoT devices from cyber threats?
- How can machine learning improve cyber deception in IoT?
- What are the benefits of using reinforcement learning (RL)-based techniques for IoT security?
- How effective is cyber deception in IoT camera security?

The overarching goal of this thesis is to develop a system that identifies vulnerabilities, captures attackers, and sustains deception for extended periods, thereby strengthening IoT security.

Objectives

General Objective

The general objective of this research is to develop a proactive security measure capable of addressing the expansion of the number of devices and adapting to new types of sophisticated vulnerabilities and attacks.

More technically, our goal is to develop a comprehensive security framework that integrates machine learning (ML) and cyber-deception techniques to protect IoT devices from cyber threats. This framework will leverage ML techniques to enable the security system to adapt to the expansion and proliferation of IoT devices.

Specific Objectives

Several specific objectives have been identified to achieve the general objective.

- Conduct a comprehensive analysis of IoT vulnerabilities and categorize their exploitation techniques;
- Develop ML models for detecting cyber threats targeting IoT devices;

- Design and implement cyber deception techniques for misleading and capturing attackers;
- Evaluate the effectiveness of the proposed security framework through simulations and real-world case studies.

These objectives will contribute to the development of a robust, adaptive, and proactive security solution tailored for IoT networks.

Key Contributions

The growing sophistication of cyber threats in IoT ecosystems demands intelligent and proactive security strategies. Traditional security mechanisms, which primarily react to attacks, often leave systems vulnerable to zero-day exploits and advanced persistent threats. Given these limitations, we propose reinforcement learning (RL) combined with ML as a compelling approach to enhancing IoT security. RL's ability to dynamically adapt and learn from evolving cyber threats enables the optimization of deception strategies and real-time threat detection. By analyzing and exploiting attack patterns, RL-based security systems can preemptively identify vulnerabilities and mitigate risks before they escalate into full-scale attacks.

This thesis aims to advance IoT security by integrating ML and cyber deception techniques to develop adaptive defense mechanisms. Specifically, our approach combines ML's predictive capabilities for threat detection with RL's adaptive learning to refine security responses over time. Additionally, we incorporate cyber deception to engage, mislead, and analyze attackers, gaining insights into their tactics while safeguarding IoT devices. This multi-layered defense strategy provides a comprehensive and dynamic security framework that evolves alongside emerging cyber threats. However, this thesis does not address adversarial attacks targeting ML models.

The key contributions of this research are as follows:

The first key contribution of this thesis is the development of a novel adaptive intelligent interaction honeypot system. This system is designed to capture and mislead attackers during the reconnaissance phase of their attacks. By simulating vulnerable IoT devices and dynamically adapting to attacker behaviors, the honeypot system effectively captures valuable data on attack methods and tactics. The main results from this contribution include the successful engagement and misleading of attackers, making it more challenging for them to distinguish between

real and decoy devices. This approach improves the security of IoT networks and provides a rich source of intelligence for improving future defensive strategies.

The second major contribution is the creation of a hybrid recommender system that uses adaptive learning to propose fake vulnerabilities alongside real ones. This system aims to deceive attackers who may bypass the initial honeypot traps or originate from within the network. By presenting a mix of genuine and deceptive vulnerabilities, the recommender system confuses and delays attackers, reducing the likelihood of successful breaches. The adaptive learning component ensures that the system evolves in response to changing attack patterns, maintaining its effectiveness over time. The key results of this contribution demonstrate the ability of the system to adaptively learn and refine its deceptive tactics, significantly enhancing the security of IoT devices.

The third important contribution is the implementation and evaluation of an augmented reality-based honeypot for IoT cameras. This honeypot creates realistic, interactive decoys that engage attackers for extended periods. By providing a visually convincing environment, these honeypots can gather detailed intelligence on attacker strategies and tools. The use of augmented reality improves the believability of the deception, making it more likely that attackers will remain engaged. The main results of this contribution highlight the practical application of cyber deception techniques in real-world scenarios, demonstrating their potential to keep attackers engaged and misled, thus protecting IoT devices from sophisticated cyber threats.

Structure of the Dissertation

This dissertation is structured to systematically explore and address the research questions and objectives outlined above.

Chapter I - Background on Internet of Things, Cyber Deception, and Machine Learning.

> This chapter lays the foundation by introducing the critical areas of cyber deception, machine learning (ML) and Internet of Things (IoT) network security. It discusses the relevance and application of these technologies in current cybersecurity frameworks, particularly focusing on the vulnerabilities and threats inherent in IoT environments and how ML can be leveraged to address these issues.

Chapter II - Related Works on Defense and Deception Strategies for IoT Security using Machine Learning.

In this chapter, we present a comprehensive review of existing research on IoT security, the role of ML in enhancing these security measures, and the implementation of cyber deception tactics. This chapter highlights the gaps in current research and technology, setting the stage for the novel contributions of this dissertation in addressing these deficiencies.

Chapter III - Contribution to Adaptive Intelligent Cyber Deception for Probing Attacks on IoT Devices.

This chapter focuses on the development of adaptive intelligent interaction honeypots designed to detect and deceive cyber attackers proactively. It details the architecture, algorithmic foundation, and implementation specifics of these honeypots, emphasizing their role in enhancing IoT security through engagement and misinformation of potential attackers.

Chapter IV - Contribution to Adaptive Learning for IoT Security: A Hybrid Recommender System Approach.

Introduces a novel hybrid recommender system that integrates real and deceptive data to mislead attackers. This chapter explains the design and operational mechanism of the system, demonstrating how adaptive learning is utilized to dynamically update the system's responses based on attacker behavior, thus enhancing the security of IoT networks.

Chapter V - Contribution to advanced Security using Augmented Reality Honeypots: A Case Study for IoT Cameras.

In this chapter, we present a practical application of the concepts discussed through a case study on augmented reality-based honeypots designed for IoT cameras. This chapter elaborates on the deployment, challenges, and effectiveness of using augmented reality to create realistic decoys that engage and deceive attackers, providing deeper insights into their tactics and strategies.

General Conclusion

We summarize the research findings and their implications for IoT security, cyber deception, and the use of machine learning. In addition, we discuss potential future directions for this research, suggesting areas for further exploration and the application of these technologies in other domains.

Appendix A - Setting Up a Honeypot on Google Cloud Platform.

We provide a step-by-step guide on how to set up a honeypot using Google Cloud Platform (GCP), configure it with a public IP address, and deploy a trained ML model for enhanced security monitoring.

Appendix B - List of Publications.

We present the scientific papers we published during this thesis.

I

CHAPTER

Background on Internet of Things, Cyber Deception, and Machine Learning

CONTENTS

I.1 - Introduction 8
I.2 - Internet of Things 9
I.3 - Cyber Deception 16
I.4 - Machine Learning 20
I.5 - Conclusion 26

I.1. Introduction

The Internet of Things (IoT) is transforming our interaction with technology, enabling devices to exchange data and perform tasks that simplify our daily lives. However, this connectivity also introduces significant security concerns, as the proliferation of IoT devices leads to increased risks of cyber threats and vulnerabilities. Traditional security measures often fall short due to the sheer diversity and resource constraints of IoT devices.

To address these challenges, cyber deception techniques have emerged as a proactive approach to cybersecurity. These techniques aim to mislead attackers, diverting them from real targets and gathering valuable intelligence. Despite their potential, creating effective deceptive data is complex and context-dependent, requiring realistic and adaptive responses to attackers' actions.

Machine learning, a branch of artificial intelligence, offers powerful tools for generating fake data and behaviors and dynamically adapting to evolving threats.

Using machine learning, it is possible to create more sophisticated and responsive cyber-deception strategies that enhance IoT security.

This chapter provides a comprehensive review of the current state of IoT security, cyber deception, and machine learning, setting the foundation for the research presented in this thesis. We will explore existing security measures for IoT, the historical and current applications of cyber deception, and the role of machine learning in revolutionizing threat detection and response. By identifying gaps and opportunities in the literature, this chapter highlights the need for advanced integrated security solutions that combine these technologies to protect the expanding IoT ecosystem.

I.2. Internet of Things

This section presents an overview of the IoT network, giving details on architecture, communication protocols, and security challenges.

I.2.1. Overview of IoT

Figure 1 – *The evolution of the internet from pre-internet to IoT [1].*

Figure 1 traces the evolution of the Internet, starting with pre-Internet technologies like telephony and SMS, evolving through the content-rich "WWW," to interactive "Web 2.0" services like e-commerce. It then progresses to the "Internet of People," highlighted by social media platforms such as Facebook and YouTube, culminating in the "Internet of Things," which connects physical devices for enhanced communication and automation. This progression demonstrates the Internet's transformation from simple communication to an advanced, interconnected network impacting all aspects of daily life.

Kevin Ashton introduced the term "Internet of Things" (IoT) during a 1999 presentation to Proctor and Gamble [9]. According to the Cambridge dictionary [10], the IoT refers to "objects with computing devices in them that can connect and exchange data using the Internet." This means that everyday objects, from household appliances to industrial machines, can communicate and perform tasks remotely

or not, improving human daily life. For example, IoT might involve a smart thermostat that can signal when the room temperature needs to be adjusted. The IEEE communication magazine defines IoT as a framework in which all things have a representation and a presence on the Internet. More specifically, the IoT aims at offering new applications and services bridging the physical and virtual worlds, in which Machine-to-Machine (M2M) communications represent the baseline communication that enables the interactions between things and applications in the cloud.

From these perspectives, we can define the Internet of Things as the interconnection of physical objects embedded with electronics, software, sensors, actuators, and other technologies and network connectivity for the purpose of connecting and exchanging data with other devices and systems over the Internet and being controlled remotely by the user.

In the context of IoT, the term 'things' generally refers to devices with distinct identities that are capable of remote sensing, actuation, and monitoring. These devices can communicate and share data with other connected systems or applications, either directly or through intermediaries. They can also gather data from other devices, process it on-site, or transmit it to centralized servers or cloud-based platforms for additional processing. Depending on temporal and spatial constraints, such as memory, processing power, communication latencies, and deadlines, tasks can be performed locally or within the broader IoT infrastructure. An IoT device can collect various types of data from onboard or attached sensors, such as temperature, humidity, and light intensity [11].

I.2.2. IoT Architectures

The architecture of IoT typically includes sensors, actuators, data processing units, and communication modules, all integrated to perform specific functions. IoT architecture consists of different layers of technologies that support the IoT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity, and configuration of IoT deployments in different scenarios. There are mainly three broad categories of architecture: the three-layer, four-layer, and five-layer architectures.

Three-layer architecture

This type of architecture consists of the perception layer, the network layer, and the application layer. The Perception layer consists of smart objects integrated with sensors, enabling real-time information collection and process-

ing. These devices can be common or simple, and can connect to the Internet for data and services. The Network layer, also known as the Transmission Layer, transmits data from the Perception layer, using technologies like 3G, WiFi, blacktooth, Infrared, and Zigbee. It manages the network, provides security, authorization, and access control, and includes firewalls and intrusion detection systems. The Application layer interfaces the IoT device with the network, processing and analyzing data collected at the Perception layer. However, processing and analyzing data generated by sensors can be overwhelming and impractical in real-time contexts [12].

Four-layer architecture

This IoT architecture model is roughly similar to the three-layer architecture, except that it has an extra layer, the middleware layer. The perception layer is the lowest layer, primarily containing sensing devices. The network layer houses network communications software and physical components, enabling device communication and data transmission. The middleware layer analyzes, stores, and processes information, providing services to lower layers and automating computations. It is also responsible for AI technologies like cloud computing and big data analysis. The application layer, based on client requests, provides intelligent services based on functional values like temperature and humidity measurements gauge. This architecture ensures high-quality services to meet client requirements. [13]

Five-layer architecture

Figure 2 illustrates a five-layer architecture model for the IoT, showing how different layers interact and function within IoT systems. In this thesis, we adopt the five-layer model, which extends the basic IoT architecture and closely resembles the four-layer model, with the addition of a business layer. This extra layer enables the data processed at the application level to be utilized for extracting business insights, forecasting future trends, and supporting decision-making processes. These decisions enhance operational efficiency, safety, cost-effectiveness, customer satisfaction, and other crucial aspects of business operations [2].

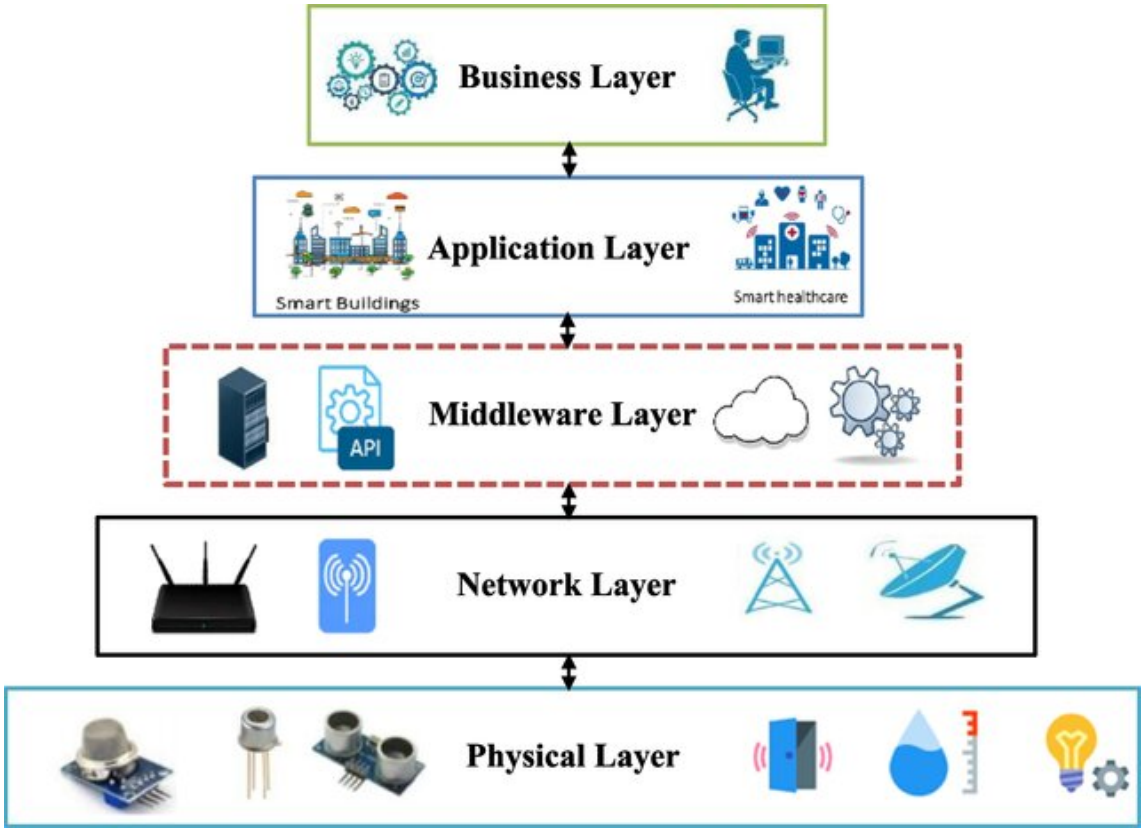


Figure 2 – *Five-layer architecture of IoT [2]*

I.2.3. IoT Technologies and Protocols

Technologies and protocols such as MQTT, CoAP, and blacktooth Low Energy (BLE) play essential roles in the operation of IoT systems. These protocols have different characteristics and applications that make them appropriate for specific IoT scenarios [14].

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol developed for networks with low bandwidth, high latency, and unreliability. It follows a publish-subscribe model, allowing devices to communicate indirectly through a broker. This model is especially beneficial in situations where devices transmit small amounts of data sporadically, such as sensor readings or status updates. Due to its simplicity and efficiency, MQTT is well-suited for IoT applications that require minimal network traffic and prioritize battery conservation.

CoAP (Constrained Application Protocol) is a web transfer protocol specifically designed for use with constrained devices and networks in the IoT. It enables low-power devices with limited processing and memory capabilities to interact with the broader internet using RESTful principles, similar to HTTP. CoAP's support for multicast, low overhead, and simplicity makes it an excellent choice

for applications like smart energy grids and home automation. Additionally, by operating over UDP (User Datagram Protocol), CoAP is effective in lossy and resource-constrained environments.

blacktooth Low Energy (BLE) is a wireless technology designed to provide low power consumption and reduced cost while maintaining a similar communication range to classic blacktooth. BLE is ideal for devices that require infrequent transmission of small data amounts, such as fitness trackers, smartwatches, and other wearable devices. Its energy-efficient design helps extend battery life, making it a popular option for consumer electronics that need to operate for long periods without frequent charging.

These technologies and protocols allow IoT devices to communicate effectively even in environments with constrained resources. By utilizing MQTT, CoAP, and BLE, IoT systems ensure reliable data transmission, smooth device interoperability, and long battery life—key factors for the success and widespread implementation of IoT solutions.

I.2.4. IoT Networks Compared to Wireless Sensor Networks and Cloud

A Wireless Sensor Network (WSN) is composed of nodes that work together to monitor and manage the surrounding environment using wireless communication. The WSN architecture consists of sensor nodes, gateways, and observers (users). Sensor fields include both gateways and sensor nodes, which are interconnected through specialized networks or the internet. In WSNs, data collection and transmission occur in four stages: data is collected, processed, packaged, and then transferred [15].

Differences between WSN and IoT include the use of wireless media, the need for a power supply, the use of sensors, the use of gateways, and the use of task managers. WSN and IoT are two fields of study that share common origins in computer networks and M2M. They differ in features, application requirements, and domains. In WSN, the security level varies according to the application, while in IoT, it is higher due to the complexity of heterogeneous things connected. Sensor networks must maintain operational robustness, even when some nodes die or leave the network. Heterogeneity is essential for interoperability among different sensor networks, while in IoT, devices' states change dynamically. WSN applications do not require object identification, but data processing involves data aggregation to reduce network congestion. In IoT, data is processed from physical/device layers and used for data analytics using Big Data techniques. In WSN, internet connection

is not essential, but in IoT, it is mandatory. Cloud computing has been introduced to address the massive amount of data generated by the IoT, which puts a huge demand on the internet infrastructure.

I.2.5. IoT Applications and Use Cases

IoT applications span multiple sectors, including healthcare, agriculture, smart homes, and industrial automation, each benefiting from the enhanced connectivity and data-driven decision-making capabilities provided by IoT.

IoT technology has become a fundamental aspect of our lives, affecting areas such as healthcare, smart water, transportation, and surveillance. It has been applied in various applications, including domestic and home automation, smart industries, smart health, and automotive and smart mobility.

Domestic and home automation involve using connected devices and appliances to make homes more comfortable, convenient, and energy-efficient. These systems can connect with central control platforms such as smartphone applications or voice-activated assistants like Amazon's Alexa or Google Assistant. In smart industries, they are used to detect gas levels and leaks in industrial settings, monitor toxic gases, and track water, oil, and gas levels in storage tanks and cisterns. In the smart health sector, connected devices and sensors play a key role in tracking health and wellness, providing real-time data to healthcare professionals, and empowering patients to actively manage their own health [16].

I.2.6. IoT Security Challenges

Securing IoT systems is a critical and complex task due to the vast number of smart devices that communicate with one another. Traditional network security relies on wired or wireless technologies to connect devices within a local area network (LAN) or wide area network (WAN). In these traditional setups, devices typically connect through a central hub or switch, facilitating data exchange between them. The key distinction between traditional networks and IoT networks lies in the variety of devices connected, which can range from home appliances to industrial machinery.

The security challenges in IoT are substantial because of the distinct characteristics of these networks. Unlike traditional networks, IoT devices generally have limited computational capabilities and power resources, making it difficult to apply robust security measures. Additionally, the wide range of devices and the absence of standardized security protocols further complicate the situation. Traditional security approaches are often not directly applicable to IoT environments, requiring

tailored security solutions that address the specific vulnerabilities of IoT devices. The integration of IoT with cloud computing also raises concerns about data privacy and integrity during both transmission and storage [7].

Due to the limited computational power, battery life, and memory capacity of IoT devices, they are particularly susceptible to attacks. As a result, IoT security solutions need to be based on lightweight algorithms. Finding the right algorithm that aligns with these constraints while providing strong security is a significant challenge. IoT devices should be designed with security built in from the outset. The key challenges in securing IoT include resource limitations, interoperability, scalability, heterogeneity, privacy protection, large data volumes, and autonomous control. Addressing these challenges requires security solutions that are lightweight, consume minimal computational power and memory, and are efficient in terms of energy consumption [17].

I.2.7. Traditional Network Security vs IoT Security

The main difference between traditional network security and IoT security lies in the entities involved in communication. Traditional network security is designed to protect networks connecting human users to computers and servers, focusing on securing data exchanges, user authentication, and access control. These networks typically involve robust hardware with significant processing power and memory, enabling the implementation of sophisticated security measures like firewalls, intrusion detection systems (IDS), and encryption protocols. The primary goal is to ensure the confidentiality, integrity, and availability of data being transmitted between users and systems [18].

In contrast, IoT security expands the scope to include machine-to-machine (M2M) communication, where numerous interconnected devices, ranging from simple sensors to complex industrial machines, communicate autonomously. This interconnectivity introduces new security challenges, as IoT devices often operate with limited computational resources and memory, making it difficult to implement traditional security mechanisms [19]. Additionally, IoT devices are frequently deployed in diverse and sometimes hostile environments, further complicating security efforts. The heterogeneity of IoT devices, coupled with the vast scale of deployments, necessitates specialized security solutions that can address the unique requirements and vulnerabilities of these systems [20].

Moreover, traditional network security primarily focuses on securing the communication channels and data at rest, while IoT security must also address the physical security of the devices themselves. IoT devices are often more acces-

sible to attackers, increasing the risk of physical tampering and hardware-based attacks [21]. Another significant difference is the lifecycle management of IoT devices. Unlike traditional IT infrastructure, IoT devices may have longer lifespans with infrequent software updates, making them susceptible to outdated security vulnerabilities [21]. The dynamic nature of IoT environments requires continuous monitoring and adaptive security measures to protect against evolving threats [22]. Therefore, IoT security demands a holistic approach that encompasses device security, network security, and data security, tailored to the specific constraints and challenges of IoT ecosystems [20].

I.3. Cyber Deception

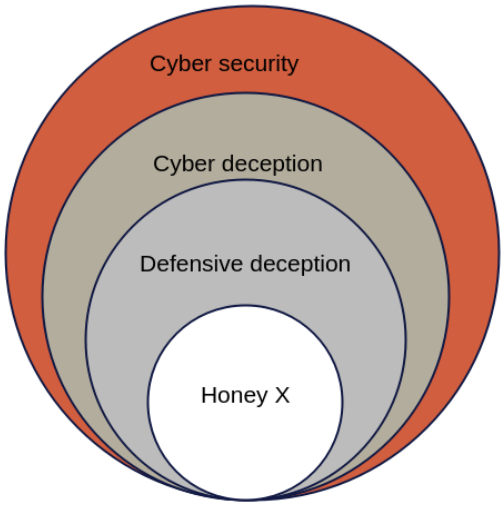


Figure 3 – *A Venn diagram depicting the nested relationship between Cyber Security, Cyber Deception, Defensive Deception, and Honey-X. Cyber Security is the broadest category, within which Cyber Deception is a subset, followed by Defensive Deception, with Honey-X being a specific implementation within Defensive Deception [3].*

The relationship among cyber security, cyber deception, defensive deception, and the concept of Honey-X can be visualized on Figure 3 to understand how these concepts are nested within each other, reflecting their specific focuses within the broader domain of cyber security. The concept of deception is highly multidisciplinary and has been studied in various domains. In this section, we explore the foundational definitions and concepts of deception, followed by the motivations for cyber deception, and the various types and properties of deceptive techniques used to enhance cybersecurity, particularly in IoT environments [23].

I.3.1. Definition and Concepts

The conventional concept of military deception refers to actions taken to intentionally mislead an enemy about one's strengths and weaknesses, intents, and tactics. The defender employs deception to manipulate the enemy's actions to advance one's mission. Defensive deception applies to a wide spectrum of interactions between an attacker and a defender under conflict situations. The concept of defensive deception initiated in military environments has been introduced to cyber domains as cyber deception. Almeshekah and Spafford expanded the idea of cyber deception, describing it as deliberate measures intended to mislead or confuse attackers, prompting them to make specific decisions that support computer security defenses [24].

Although deception spans across multiple disciplines, the common understanding is that it involves misleading an entity into adopting a false belief, which in turn influences its behavior. When deception is successfully carried out, the target reacts in a less effective manner, benefiting the deceiver and helping them achieve their intended objective. Rowe and Rrushi [25] emphasized 'manipulation' as the core concept of deception, where deception encourages a deceivee to do something a deceiver wants or discourages the deceivee from doing something the deceiver does not want.

I.3.2. Multidisciplinary Perspectives on Deception

Philosophy: In philosophy, deception is often defined as an intentional act to mislead an entity into believing a false belief. The core aspects of deception in philosophy lie in the intentionality and the creation of false beliefs [26].

Behavioral Science: Behavioral scientists, including biologists, ecologists, neuroscientists, and medical scientists, investigate deception through the behaviors of animals or humans. They focus on functional deception, which misleads the actions of others, and intentional deception, which involves intentional states like beliefs and desires guiding behavior [27].

Sociology: Sociological research on deception examines its effects in various social contexts, both positive and negative, and its use as a relational or marketing strategy [28].

Psychology: Psychologists define deception as behavior that provides information to mislead subjects. Research in this field often focuses on identifying cues for deception, psychological symptoms of self-deception, and individual differences in deceptive behavior [29].

Public Relations: In public relations, self-deception is studied as a strategic solution to resolve internal or external crises, often by attributing problems to individuals or victims to avoid disastrous impacts on organizations [30].

Communications and Linguistics: Deception research in communications and linguistics aims to identify verbal or non-verbal indicators of deceptive communication [31].

Command and Control: In the military domain, deception refers to planned maneuvers that reveal false information and hide the truth to mislead the enemy. Military deception involves many individuals or organizations and takes place over long periods [32].

Computing and Engineering: In computing and engineering, deception techniques enhance security by diverting and misleading potential attackers, detecting their presence, and gaining valuable insights into their tactics and motives. Techniques such as honeypots, decoys, and misinformation help protect systems against evolving cyber threats [33].

I.3.3. Motivations for Cyber Deception

Cyber deception aims to increase the cost and complexity for attackers, divert their attention, and delay their progress. Deception techniques can be effective in the early detection and mitigation of cyber threats, reducing the impact of attacks. The use of deceptive measures also assists in gathering intelligence about attackers and their methods.

I.3.4. Types and Properties of Cyber Deception

Conceptual Deception Tactics

- **Masking**: Hiding certain information or data in the background.
- **Repackaging**: Hiding a real object as something else to conceal its true nature.
- **Dazzling**: Overshadowing a real object to make it hard to detect.
- **Mimicking**: Imitating aspects of a real object to make honeypots appear as real nodes or interfaces.
- **Inventing**: Creating new fake objects to lure attackers.
- **Decoying**: Attracting an attacker's attention away from critical assets.
- **Bait**: Using truthful but low-grade information to lure attackers.

Expected Effects of Defensive Deception

- **Hiding**: Changing the display of a real object to something else.
- **Luring**: Demonstrating fake objects that look real to attackers.
- **Misleading**: Leading an attacker to choose suboptimal actions.
- **Blending**: Making real objects blend into the background to avoid detection.
- **Confusing**: Creating perceived uncertainty to confuse attackers.

Ultimate Goals of Defensive Deception

The ultimate goals are to either protect assets or detect attackers. Deception by hiding often aims to protect assets, while using false objects or information aims to catch attackers.

Activeness of Defensive Deception

Deception can be passive, involving hiding valuable assets, or active, involving the use of fake information to mislead attackers. The goal can be either protecting assets or detecting attackers.

I.3.5. Distinctions between Defensive Deception and Other Defense Techniques

Moving Target Defense (MTD): MTD aims to increase confusion or uncertainty for attackers but does not use false information to mislead attackers. Defensive deception, however, often involves using false objects or information to manipulate attackers' perceptions [34].

Obfuscation: Obfuscation techniques are used by both attackers and defenders to obscure code or data. While obfuscation directly achieves security goals, defensive deception involves cognitive aspects of an attacker's perception, making it distinct but often overlapping with obfuscation [35].

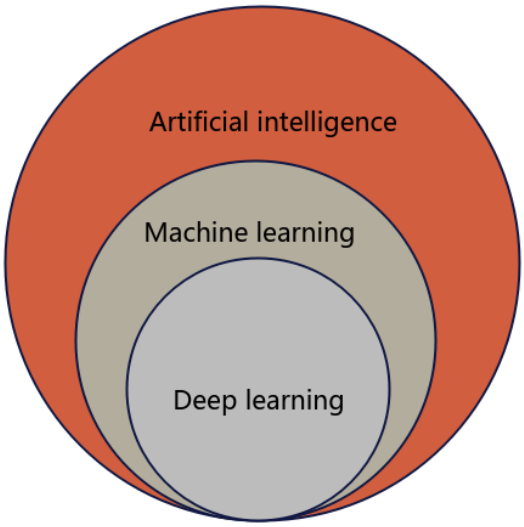


Figure 4 – *A Venn diagram illustrating the hierarchical relationship between Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL). AI encompasses the broadest field, within which ML is a subset, and DL is a further subset within ML [4].*

I.4. Machine Learning

The relationship between artificial intelligence (AI), machine learning (ML), and deep learning (DL) can be visualized on Figure 4 to understand their hierarchical structure and how they encompass each other. Machine learning (ML) is a subset of artificial intelligence that involves training algorithms to learn from data and make predictions or decisions based on that data. ML encompasses a range of algorithms and models, including supervised learning, unsupervised learning, and reinforcement learning. Each type of learning addresses different aspects of data analysis and prediction.

I.4.1. Overview of Machine Learning

For centuries, inventors have envisioned machines capable of thinking. Ancient Greek mythology featured stories of intelligent objects, like statues that could move and tables that delivered food and drink on command. Even before programmable computers were built, there was speculation about their potential to become intelligent. Today, artificial intelligence (AI) has grown into a thriving field with numerous practical uses and ongoing research. We now rely on intelligent software to automate tasks, interpret speech and images, assist in medical diagnoses, and aid in scientific research.

In the initial stages of AI, the field quickly addressed and solved challenges that are intellectually challenging for humans but relatively simple for comput-

ers—problems that can be defined by a set of formal, mathematical rules. The true challenge to AI proved to be solving the tasks that are easy for people to perform but hard for people to describe formally—problems that we solve intuitively, that feel automatic, like recognizing spoken words or faces in images.

I.4.2. Types of Learning

One widely accepted definition of learning in computer programs states that "A computer program is considered to learn from experience E regarding a certain set of tasks T and performance measure P, if its ability to perform tasks in T, evaluated by P, improves with experience E" [36]. Machine learning is generally divided into three primary categories: supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning

Supervised learning refers to the process of training a model using labeled data, where the correct outcome for each data point is already provided. This approach is useful for tasks like classification and regression. For example, a supervised learning model can be trained on a dataset of network traffic labeled as either normal or malicious. The model can then classify new, unseen traffic as either benign or suspicious based on its training. Most supervised learning algorithms estimate a probability distribution $p(y|x)$. For instance, logistic regression uses the logistic sigmoid function to squash the output of a linear function into the interval (0, 1) and interpret that value as a probability. There is no closed-form solution for its optimal weights, so we maximize the log-likelihood using gradient descent [37].

Unsupervised Learning

Unsupervised learning is used with unlabeled data to identify hidden patterns or intrinsic structures in the input data. Techniques such as clustering can group similar data points together, which can help in identifying unusual network activities that deviate from the norm, potentially signaling a security breach. [37].

Reinforcement Learning

Reinforcement Learning (RL) is an ML approach where an agent learns to make decisions by taking actions within an environment to maximize overall

rewards. Important elements in RL include the agent, environment, actions, rewards, and policy. The agent engages with the environment, gets feedback through rewards or penalties, and adjusts its policy to enhance future decision-making. RL is particularly useful in dynamic and complex environments, such as cybersecurity, where adaptive and intelligent responses are required to counter evolving threats[37].

RL problems are typically modeled as Markov Decision Processes (MDPs), defined by a tuple (S, A, P, R, γ), where S is a set of states, A is a set of actions, P represents state transition probabilities, R is the reward function, and γ is the discount factor. The goal of the RL agent is to find an optimal policy π^* that maximizes the expected cumulative reward [38]. This is expressed as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid \pi\right] \tag{I.1}$$

RL can be applied in cybersecurity to develop adaptive defenses that learn from interactions with attackers. For instance, an RL-based honeypot can dynamically adjust its behavior to keep attackers engaged and gather valuable intelligence. Implementing RL in real-world systems involves challenges such as balancing exploration and exploitation, ensuring scalability to handle large state and action spaces, and maintaining robustness against changes and uncertainties in the environment.

The application of reinforcement learning in IoT security has gained traction, offering adaptive and self-learning capabilities to combat evolving security challenges. Reinforcement learning algorithms enable IoT systems to dynamically adjust their security measures based on real-time feedback and environmental changes, thereby improving their overall security posture.

I.4.3. Deep Learning

Deep learning, a specialized branch of ML, employs neural networks with multiple layers to model complex patterns in large datasets [39]. This approach has been particularly successful in image recognition, natural language processing, and autonomous systems. Neural networks, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown great promise in handling diverse and unstructured data types common in cybersecurity [40].

Transformer-based architecture (BERT)

BERT (Bidirectional Encoder Representations from Transformers) [41] is part of the transfer learning methods family, where a model is initially pre-trained on broad tasks and later fine-tuned for specific target tasks. Transfer learning has shown advantages across a wide range of tasks.

BERT is a deep learning model based on the Transformer architecture, introduced by the transformer. It relies entirely on attention mechanisms to draw global dependencies between input and output [42]. This architecture allows for parallelization, significantly improving training times over previous sequential models like RNNs and LSTMs.

The core component of the Transformer is the self-attention mechanism, which computes a representation of the input sequence by relating different positions within the sequence. The attention mechanism is defined as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{I.2}$$

where Q (queries), K (keys), and V (values) are matrices derived from the input embeddings, and d_k is the dimension of the keys. This allows the model to focus on different parts of the input for each output element.

BERT utilizes a bidirectional approach, meaning it considers the entire context of a word by looking at both the preceding and following words (left and right context) in all layers. This bidirectional nature is achieved by training BERT on two unsupervised tasks:

1. Masked Language Model (MLM): Randomly masks some of the words in the input and predicts them.
2. Next Sentence Prediction (NSP): Predicts whether two given sentences are consecutive in the original text.

The training of BERT is designed to capture deep contextual representations, which can then be fine-tuned on downstream tasks such as text classification, named entity recognition, and question answering by adding task-specific layers and fine-tuning the entire architecture on annotated data.

Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) [43]. consist of two competing neural networks, a generator and a discriminator, which are trained simultaneously through adversarial learning. The generator attempts to produce

realistic synthetic data, while the discriminator evaluates the authenticity of the data, distinguishing between real and fake samples.

Mathematically, the training process of GANs is formulated as a minimax game with the following value function $V(G,D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))], \quad \text{(I.3)}$$

where:

- $D(x)$ represents the probability that x came from the real data distribution $p_{\text{data}}(x)$.
- $G(z)$ represents the generator's output when given a noise vector z sampled from a prior distribution $p_z(z)$.

The generator G aims to maximize the probability of the discriminator making a mistake, while the discriminator D aims to correctly classify real and fake data.

In IoT security, GANs can be applied to generate synthetic attack scenarios to train and evaluate security systems. For instance, a GAN can be used to simulate network traffic patterns that resemble those of real-world attacks, thereby improving the robustness of intrusion detection systems.

Recommendation Systems

Recommendation systems are algorithms designed to suggest relevant items to users based on their preferences and behaviors . These systems leverage various techniques, including collaborative filtering, content-based filtering, and hybrid methods [44].

1. **Collaborative Filtering**: This method predicts a user's interests by collecting preferences from many users. Two main approaches are used:
   - **User-based collaborative filtering**: Predicts a user's rating for an item based on the ratings of similar users.
   - **Item-based collaborative filtering**: Predicts a user's rating for an item based on the user's ratings of similar items.

Mathematically, the similarity between users or items can be measured using metrics like cosine similarity:

$$\text{cosine-similarity}(A,B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{I.4}$$

2. **Content-Based Filtering**: This method recommends items similar to those the user has liked in the past. It relies on the attributes of items and the user's profile.

3. **Hybrid Methods**: Combine collaborative and content-based filtering to leverage the strengths of both approaches and mitigate their weaknesses.

In cybersecurity, recommendation systems can suggest appropriate security measures, configurations, or actions based on the current threat landscape and specific characteristics of the IoT environment. By analyzing historical data and real-time analytics, recommendation systems can improve the overall security posture of IoT systems. For example, a hybrid recommender system can suggest fake vulnerabilities alongside real ones to deceive attackers and delay their progress.

Augmented Reality (AR)

Augmented Reality (AR) integrates digital information with the user's environment in real-time. AR overlays virtual objects onto the real world, enhancing the user's perception and interaction with their environment [45].

In cybersecurity, AR can be used to create immersive and interactive deception environments that engage attackers and collect valuable intelligence. For example, an AR-based honeypot can simulate a realistic IoT environment, making it difficult for attackers to distinguish between real and fake devices. By engaging attackers in an AR environment, security teams can monitor their behavior and collect data on their tactics, techniques, and procedures.

To implement AR in cybersecurity, we use computer vision and image recognition techniques to accurately overlay digital information onto the real world. Key components include:

- **Cameras and sensors**: Capture the real-world environment and track the user's movements.
- **Display devices**: Provide visual feedback to the user, such as AR glasses or smartphone screens.

- **AR software**: Processes the captured data and generates the augmented content.

By creating realistic decoys, AR-based honeypots can keep attackers engaged for extended periods, providing valuable insights into their behavior and improving the overall security of IoT ecosystems.

I.4.4. Applications of ML in Cybersecurity

In the context of cybersecurity, ML plays a pivotal role in enhancing threat detection by analyzing vast amounts of data to identify patterns and anomalies that may indicate malicious activities [46]. Traditional security measures often rely on predefined rules and signatures to detect threats, which can be bypassed by sophisticated attackers who use novel techniques. ML algorithms, however, can learn from historical data and adapt to new and emerging threats, making them more effective in identifying previously unknown attack vectors. Anomaly detection is one such technique where ML models are trained to recognize normal behavior within a network or system. When deviations from this norm are detected, these anomalies can be flagged for further investigation, potentially uncovering cyber threats that might have gone unnoticed with conventional methods [47].

In addition, ML can automate the identification of anomalies and streamline response strategies, thus significantly reducing the time and effort required for cybersecurity teams to respond to incidents. Predictive analytics, powered by ML, can forecast potential security breaches by analyzing trends and patterns in data, allowing organizations to take proactive measures to fortify their defenses. Automated threat intelligence systems use ML to gather, analyze, and disseminate information about threats in real-time, providing security professionals with up-to-date insights to make informed decisions. These applications of ML not only enhance the efficiency and effectiveness of cybersecurity operations but also help in building more resilient and adaptive security infrastructures capable of withstanding sophisticated cyber attacks.

I.5. Conclusion

In this chapter, we provided a comprehensive overview of the foundational concepts essential for understanding IoT security, cyber deception, and machine learning (ML). We began by exploring the definition, evolution, architecture, technologies, and protocols of the Internet of Things (IoT), highlighting the significant

security challenges it poses. We then examined the concept of cyber deception, emphasizing its role in enhancing cybersecurity by proactively misleading and confusing attackers. Finally, we introduced machine learning, detailing its types and applications in cybersecurity, particularly for threat detection, anomaly identification, and response automation. By integrating machine learning techniques with cyber deception strategies, we can create autonomous and adaptive security mechanisms that significantly enhance the resilience of IoT ecosystems against sophisticated cyber threats. This chapter has laid the groundwork for the subsequent exploration of advanced security solutions that leverage these technologies to address the evolving threats in the IoT landscape. As IoT continues to grow, it is imperative for researchers and practitioners to stay informed about emerging trends in cyber deception and machine learning to effectively safeguard IoT ecosystems.

While integrating ML with cyber deception offers promising security enhancements, it also presents several challenges. One major issue is the complexity of designing deception strategies that remain effective against increasingly sophisticated attackers. Attackers continuously adapt to security measures, meaning that deception models must evolve dynamically to maintain their effectiveness. Additionally, ML models used for deception can be vulnerable to adversarial attacks, where attackers manipulate input data to deceive or bypass security mechanisms.

Another significant challenge is the computational and resource constraints of IoT devices. Many IoT systems operate with limited processing power, memory, and energy, making it difficult to deploy advanced ML models or real-time deception techniques. Balancing security effectiveness with resource efficiency remains an open research question.

Furthermore, ethical and legal considerations must be taken into account when implementing cyber deception techniques. Using deceptive tactics to engage attackers raises concerns about unintended consequences, such as legal implications of monitoring and misleading intruders. Organizations deploying these measures must navigate regulatory frameworks and ensure that deception techniques align with ethical cybersecurity practices.

Despite these challenges, the integration of ML with cyber deception remains a promising avenue for strengthening IoT security. By addressing these limitations and refining adaptive deception strategies, future research can enhance the resilience of IoT ecosystems against evolving cyber threats. This chapter has laid the groundwork for exploring advanced security solutions that leverage these technologies, which will be further examined in the subsequent chapters.

II

CHAPTER

Related Works on Defense Strategies for IoT Security using Machine Learning

CONTENTS

II.1 - Introduction 28
II.2 - Overview of IoT Security 29
II.3 - Security Threats on IoT Devices 33
II.4 - Machine Learning for IoT Security 37
II.5 - Cyber Deception for IoT Security 40
II.6 - Conclusion 46

II.1. Introduction

The rapid proliferation of IoT devices has transformed various sectors by enabling seamless connectivity and data exchange. Despite its widespread use, this has brought about notable security issues, as IoT devices frequently function in varied and resource-limited settings, leaving them exposed to cyberattacks. This chapter provides a comprehensive review of the current state of research in the fields of IoT security, cyber deception, and machine learning. We explore the various challenges, methodologies, and technologies employed to enhance the security of IoT environments. The integration of IoT devices into everyday life and critical infrastructure has resulted in a complex landscape where traditional security measures often fall short. Consequently, there is an urgent need for innovative approaches that can proactively detect, mitigate, and respond to evolving cyber threats.

The first section of this chapter delves into IoT security, providing an overview of the current landscape, identifying key security and privacy issues, and discussing

advanced authentication methods, such as biometric systems. We also examine the role of blockchain technology in improving IoT security, as well as the layered security approaches necessary to protect the diverse components of IoT systems. The second section focuses on the application of machine learning in IoT security. ML methods have shown potential in recognizing anomalies, detecting intrusions, and forecasting possible threats in real time. Using data-driven models, machine learning can provide adaptive and scalable security solutions tailored to the unique characteristics of IoT environments. We discuss various ML approaches, including supervised, unsupervised, and reinforcement learning, highlighting their potential to fortify IoT ecosystems against cyber threats.

In the third section, we explore the concept of cyber deception as a proactive defense mechanism. Cyber deception involves misleading attackers by deploying deceptive elements such as honeypots, which are designed to gather intelligence on attacker tactics and delay their progress. We examine different types of honeypots, their interaction levels, and the strategic deployment of these tools to enhance IoT security. Additionally, we discuss the integration of cyber deception with machine learning techniques to create intelligent and adaptive honeypot systems. Furthermore, we highlight the role of recommender systems and augmented reality in improving IoT security. Recommender systems can provide personalized and adaptive security recommendations, while augmented reality can create immersive environments that engage attackers and gather valuable intelligence. The chapter concludes with a critical analysis of the existing literature, identifying gaps and areas for future research. By integrating the insights from IoT security, machine learning, and cyber deception, we aim to present a holistic approach to safeguarding IoT systems against emerging cyber threats.

II.2. Overview of IoT Security

IoT has transformed the way devices connect and communicate with each other, offering huge opportunities for automation, efficiency, and convenience. However, this interconnection also presents significant security challenges. The lack of robust authentication mechanisms and encryption protocols, along with resource constraints, makes IoT devices vulnerable to cyber attacks. These vulnerabilities make IoT environments attractive targets for malicious actors, who can exploit the interconnected network to deploy malware and compromise the integrity, confidentiality, and availability of sensitive data and systems. To mitigate these security risks, organizations and individuals need to prioritize IoT security measures.

These security measures should include the implementation of strong authentication protocols, encryption algorithms, and access controls to protect IoT devices and networks. In addition, regular software updates and patches should be applied to address known security vulnerabilities. Furthermore, organizations should invest in network monitoring and anomaly detection systems to identify suspicious activities or attempts at unauthorized access.

II.2.1. Identification of Security and Privacy Issues

One significant area of concern in IoT security is the identification of security and privacy issues within IoT systems. The rapid proliferation of IoT devices has introduced a vast array of security challenges, primarily due to the inherent complexity and heterogeneity of these devices. Each IoT device often comes with its own set of hardware specifications, software configurations, and communication protocols, which collectively contribute to a diverse and fragmented IoT ecosystem. Securing these diverse components is a formidable task, as it requires tailored security measures that can effectively address the specific vulnerabilities of each device. Furthermore, the resource-constrained nature of many IoT devices, such as limited processing power and memory, further complicates the implementation of robust security mechanisms. These limitations make it difficult to apply traditional security solutions, necessitating the development of lightweight and efficient security algorithms that can operate within these constraints.

Moreover, privacy concerns are equally critical in the context of IoT security. The widespread deployment of IoT devices often involves the collection and processing of vast amounts of sensitive data, ranging from personal information to critical industrial data. Ensuring the privacy of this data requires implementing robust privacy models that can protect user information across different layers of IoT applications. This involves not only securing the data during transmission and storage but also ensuring that data processing and access control mechanisms are in place to prevent unauthorized access and misuse. Effective security solutions must therefore incorporate privacy-preserving techniques, such as encryption, anonymization, and access control policies, to mitigate the risks associated with data breaches and unauthorized data access. By addressing both technical and privacy concerns, a holistic approach to IoT security can be achieved, safeguarding the integrity, confidentiality, and availability of IoT systems and the data they handle [48].

II.2.2. Biometric Authentication

Integrating biometrics offers a way to improve both the security and usability of IoT devices. Biometric authentication, which includes techniques like fingerprint scanning, facial recognition, and voice identification, relies on distinctive physiological and behavioral traits to confirm an individual's identity. Unlike conventional password-based systems, which are vulnerable to being forgotten, guessed, or stolen, biometric authentication provides enhanced security because biometric traits are unique and generally permanent. For example, fingerprints and facial features are unique to each person, making it difficult to replicate or forge, making biometric authentication a strong option for safeguarding IoT devices.

Moreover, biometric authentication significantly enhances user convenience and experience. Traditional authentication methods often require users to remember complex passwords or carry tokens, which can be cumbersome and inconvenient. In contrast, biometric systems enable seamless and quick authentication processes, where users can simply use their fingerprint, face, or voice to gain access. This ease of use is particularly beneficial in the context of IoT, where devices are often designed for quick and effortless interactions. By reducing the reliance on passwords and other traditional methods, biometrics can streamline the user experience while maintaining high security standards.

However, the deployment of biometric authentication in IoT devices is not without challenges. One of the primary concerns is the privacy and security of the biometric data itself. Biometric information, once compromised, cannot be changed like a password. Therefore, it is crucial to implement strong encryption and secure storage solutions to protect biometric data from unauthorized access and breaches. In addition, there are concerns about the accuracy and reliability of biometric systems, as factors such as lighting conditions, physical changes, and environmental noise can impact the performance of biometric sensors. To address these challenges, ongoing research and development are focused on improving the robustness and resilience of biometric systems, ensuring they can operate effectively in diverse real-world conditions [49].

Furthermore, the integration of biometrics into IoT devices must consider the potential implications for user privacy. Biometric systems inherently involve the collection and processing of sensitive personal data, raising concerns about surveillance and data misuse. It is essential to establish clear policies and regulations that govern the use of biometric data, ensuring transparency and user consent. By addressing these ethical and privacy concerns, biometric authentication can be effectively integrated into IoT devices, providing a secure, user-friendly, and

privacy-conscious solution to some of the prevalent security challenges in the IoT ecosystem.

II.2.3. Blockchain for IoT Security

The use of blockchain technology has been proposed as a means to enhance IoT security. Blockchain, a decentralized and distributed ledger technology, can potentially address the myriad of security challenges introduced by IoT network communications. In IoT environments, where numerous devices constantly exchange data, ensuring the authenticity and integrity of these communications is paramount. Blockchain's inherent characteristics, such as its decentralized nature and cryptographic security, make it an ideal candidate for securing IoT operations. By decentralizing the control and management of data, blockchain eliminates single points of failure, reducing the risk of data tampering and unauthorized access. Furthermore, the transparency and traceability provided by blockchain ensure that all transactions are recorded immutably, enhancing trust and accountability within the IoT ecosystem [50].

The integration of blockchain with IoT can also facilitate secure device authentication and management. Each IoT device can be registered on the blockchain, providing a verifiable identity that cannot be easily spoofed or tampered with. This ensures that only authorized devices can participate in the network, mitigating the risk of malicious entities gaining access. Additionally, smart contracts, which are self-executing contracts with the terms of the agreement directly written into code, can automate and enforce security policies across IoT devices. For instance, a smart contract could automatically revoke access for devices that exhibit suspicious behavior, thereby enhancing the overall security posture. However, while the benefits are significant, there are challenges to be addressed, such as the scalability of blockchain networks to handle the vast amount of data generated by IoT devices and the computational overhead associated with blockchain operations. Ongoing research is focused on optimizing blockchain protocols to better suit the resource-constrained nature of IoT devices, ensuring that the integration of blockchain can be both secure and efficient [51].

II.2.4. Layered Security Approach

Furthermore, the security requirements for IoT span various layers, including sensing, network, service, and application interface layers. Each of these layers has unique security challenges that require specific protective measures. For example, the sensing layer is responsible for data collection from various sensors and

devices, making it crucial to protect the integrity and confidentiality of the data at this initial stage. Techniques such as data encryption, secure boot mechanisms, and tamper-resistant hardware are essential to ensure that the collected data is accurate and unaltered. Additionally, access control mechanisms must be implemented to ensure that only authorized devices and users can interact with the sensors and collect data.

The network layer, on the other hand, is primarily concerned with the transmission of data between devices and systems. Ensuring the security of this layer involves protecting data in transit from eavesdropping, tampering, and other malicious activities. This can be achieved through the use of secure communication protocols, such as Transport Layer Security (TLS) and Internet Protocol Security (IPsec), which provide encryption and integrity checks for data packets. Additionally, implementing network segmentation and firewalls can help to isolate and protect critical parts of the network from potential attacks. The service layer focuses on the management and orchestration of IoT services, requiring robust authentication and authorization mechanisms to prevent unauthorized access and ensure the integrity of service operations. Finally, the application interface layer, which interacts directly with end-users, must incorporate secure coding practices and regular vulnerability assessments to protect against exploits and ensure a secure user experience [52]. The security measures of each layer collectively contribute to a comprehensive and resilient IoT security framework, addressing the complex and multifaceted nature of IoT ecosystems.

II.3. Security Threats on IoT Devices

IoT devices are exposed to a wide range of security threats due to their inherent vulnerabilities and widespread connectivity. Figure 5 depicted a classification of IoT attacks based on the layer they operate. Below is a list of common security threats that affect IoT devices:

- **Unauthorized Access**: Attackers gain unauthorized access to IoT devices to control or manipulate them. This is often due to weak authentication mechanisms and default passwords [53].
- **Botnets**: Compromised IoT devices are co-opted into botnets, which are used to launch large-scale distributed denial-of-service (DDoS) attacks [54].

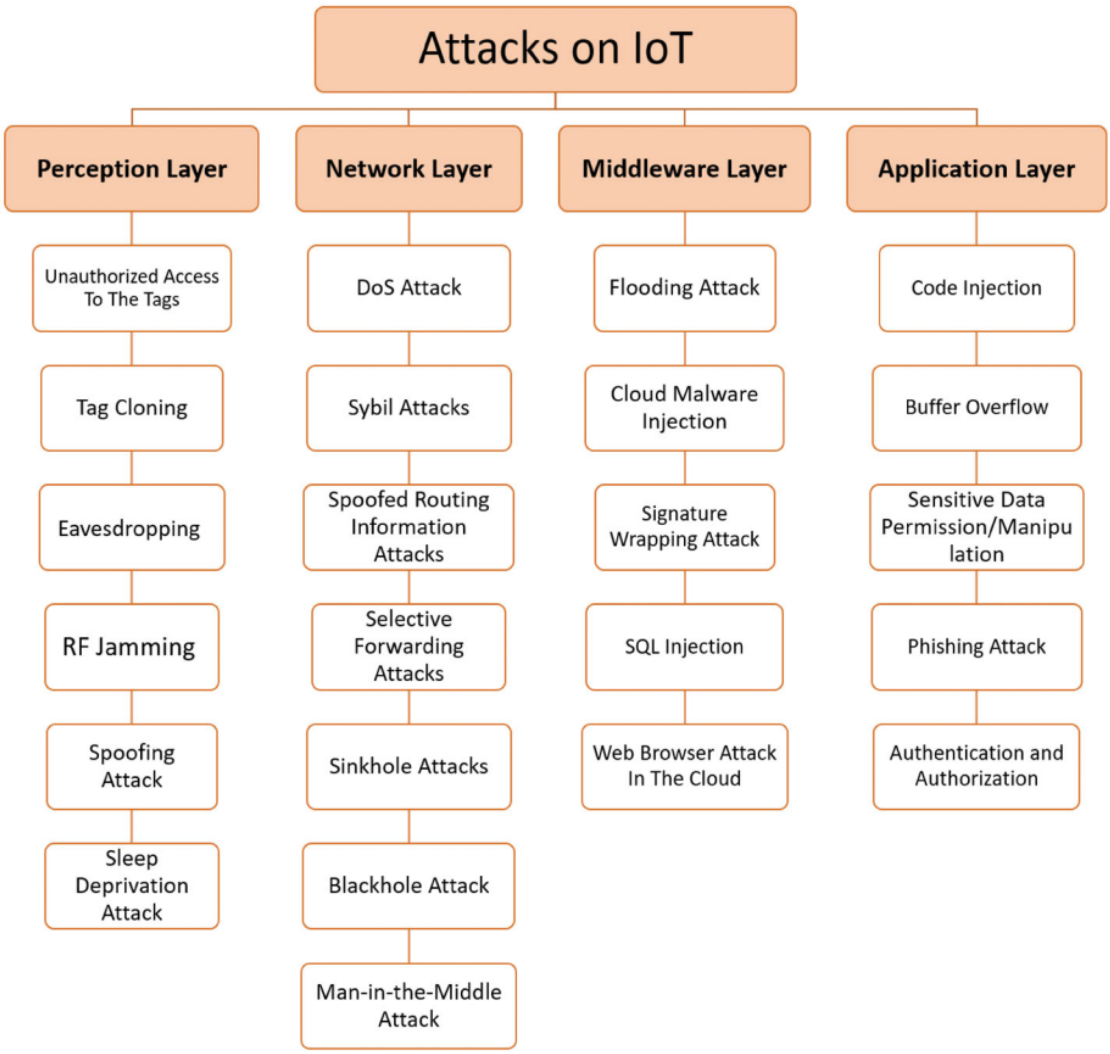


Figure 5 – *IoT attack classification [1].*

- **Man-in-the-Middle (MitM) Attacks**: Attackers intercept and alter communications between IoT devices and their controllers, leading to data breaches and unauthorized control [55].
- **Data Exfiltration**: Sensitive data is extracted from IoT devices, often exploiting vulnerabilities in device firmware or network communication protocols [56].
- **Firmware Attacks**: Malicious firmware updates are used to compromise IoT devices, providing persistent control to attackers [57].
- **Ransomware**: Attackers encrypt data on IoT devices or lock them, demanding a ransom to restore functionality [58].

- **Physical Attacks**: Physical access to IoT devices allows attackers to tamper with hardware and gain control or extract sensitive information [59].
- **Denial of Service (DoS)**: Overloading IoT devices with traffic to render them non-functional [54, 60, 61].
- **Reconnaissance Attacks**: Attackers perform network scans to gather information on IoT devices, identifying vulnerabilities to exploit [62].
- **Side-Channel Attacks**: Attacks that exploit information leaked from physical characteristics of the device, such as power consumption or electromagnetic emissions [63].

In this thesis, we focus on three primary security threats that pose significant risks to IoT devices: reconnaissance, vulnerability exploitation, and data exfiltration.

II.3.1. Reconnaissance Attacks

Reconnaissance attacks are often the initial phase of a cyber attack in which attackers perform network scans to gather information about the target network and its devices. This information includes active devices, open ports, and running services, which helps attackers identify potential vulnerabilities to exploit [62]. According to Rana et al. [64], reconnaissance attacks are crucial in hacking IoT devices and facilitating cybercrime. The study highlights how simple port scans using tools like nmap can reveal significant information about target devices, making them vulnerable to further exploitation [65].

Sharmin and Kiekintveld [66] explore the analysis of reconnaissance attacks on IoT devices by utilizing time-based features and flag attributes. Their methodology improves the accuracy and efficiency of predicting such attacks, demonstrating significant progress in the evaluation of the security of IoT devices.

Keshavamurthy and Kulkarni [67] concentrate on the early detection of probing attacks by examining performance and traffic characteristics. Their study highlights promising outcomes in automating early reconnaissance detection and underscores the importance of proactive strategies to protect IoT devices.

II.3.2. Vulnerability Exploitation

Vulnerability exploitation occurs when attackers take advantage of identified weaknesses in IoT devices, such as outdated firmware, unpatched software, or weak passwords, to gain unauthorized access and control [54]. Exploiting these

vulnerabilities can lead to unauthorized data access, device manipulation, and service disruptions. Exploiting vulnerabilities in IoT devices is a common strategy for attackers. Lightbody et al. [68] propose using non-intrusive side channel power data for intrusion detection, particularly focusing on reconnaissance and brute force attacks. Their study provides insights into how side-channel power signatures can be used to understand and reverse engineer attacks.

D'Orazio et al. [69] examine data exfiltration from IoT devices, particularly iOS devices, by exploiting vulnerabilities in pairing mode. Their case studies demonstrate how attackers can exfiltrate data using legitimate libraries and command-line tools, highlighting the critical need for robust security measures.

Uroz and Rodrguez [70] explore the use of IoT protocols for data exfiltration, reviewing protocols such as CoAP, MQTT, and AMQP. Their findings indicate that MQTT and AMQP are particularly effective for data exfiltration due to their widespread use in IoT environments.

II.3.3. Data Exfiltration

Data exfiltration involves the unauthorized extraction of sensitive data from IoT devices. Attackers exploit vulnerabilities in device firmware or network communication protocols to access and steal data, which can include personal information, usage patterns, and other sensitive details [56]. These attacks can lead to significant privacy breaches and potential misuse of the stolen data. For example, an attacker might use exfiltrated data to gain deeper insight into user behaviors, preferences, and even location, which could then be used for further malicious activities such as identity theft or targeted phishing attacks [57]. The risk of data exfiltration is increased by the often inadequate security measures implemented in many IoT devices, which are designed primarily for functionality rather than robust security [55].

The methods used for data exfiltration can vary widely, from simple network packet sniffing to more sophisticated techniques like DNS tunneling or exploiting side-channel vulnerabilities [63]. Attackers may use malware to establish a persistent presence on the device, allowing them to continuously exfiltrate data over an extended period. Furthermore, the decentralized nature of IoT networks can make it difficult to monitor and detect unauthorized data flows effectively. Addressing data exfiltration requires a multifaceted approach that includes securing firmware updates, encrypting data transmissions, and implementing advanced anomaly detection systems to identify suspicious activities [54, 62]. Furthermore, regular se-

curity audits and updates can help mitigate vulnerabilities and protect sensitive data from being exfiltrated [58].

Reconnaissance, vulnerability exploitation, and data exfiltration pose significant risks to the integrity and availability of IoT systems. Continuous research and development of advanced detection and prevention mechanisms are crucial in safeguarding these devices from malicious activities.

II.4. Machine Learning for IoT Security

Table I – *Applications of ML in Cybersecurity*

Application	Description	References
Threat Detection & Intrusion Prevention	ML models analyze network traffic to identify malicious activities.	[71]
Anomaly Detection	Detects deviations from normal behavior, identifying potential cyber threats.	[72]
Cyber Deception & Honeypots	Uses ML to create intelligent honeypots that adapt to attacker behavior.	[73, 74, 75]
Automated Incident Response	ML-driven systems automate threat mitigation and response strategies.	[76]
Malware Analysis	Uses ML classifiers to detect and categorize malware samples.	[77]
Phishing Detection	Identifies phishing emails and URLs using natural language processing (NLP).	[78]

ML techniques have proven to be a promising method for improving the security of IoT devices. Several studies have explored the potential of machine learning algorithms in securing IoT systems [79]. These techniques encompass various aspects of IoT security, including anomaly detection, intrusion detection, predictive analytics, and malware detection [80, 81]. Researchers are utilizing ML approaches, including supervised learning, unsupervised learning, and RL, to strengthen IoT environments against cyber threats [82, 83, 84]. The utilization of machine learning in IoT security is important due to the evolving threat landscape and the need for adaptive security measures [81, 85]. ML techniques can be leveraged for tasks such as user authentication, malware detection, role-based access control, and secure data downloading, contributing to bolstering IoT security measures [86]. By analyzing patterns and anomalies in the data, ML algorithms can identify potential security threats in real-time, offering a proactive approach to IoT security. Table I summarises the application of ML in cybersecurity.

Machine learning offers a dynamic and intelligent means to address security concerns in IoT environments. With the ability to process vast amounts of data and identify patterns that may indicate security breaches, ML algorithms provide an essential tool for enhancing the security of IoT devices. These capabilities are crucial in today's rapidly changing threat landscape, where traditional security measures often fall short. By incorporating machine learning into security protocols, organizations can develop more robust and adaptive defense mechanisms, ensuring the continued safety and functionality of their IoT systems.

II.4.1. Anomaly Detection

A major benefit of using ML for IoT security is its ability to identify anomalies and potential security threats in real time [87, 88]. By analyzing data from IoT networks, machine learning algorithms can identify abnormal patterns indicative of malicious activities, such as Distributed Denial of Service (DDoS) attacks and sensor tampering [89]. This real-time detection capability is crucial to preventing significant damage to IoT systems, as it allows immediate response to potential threats.

Anomaly detection in IoT systems involves monitoring network traffic and device behavior to identify deviations from established norms. ML algorithms can analyze historical data to create a baseline of typical activity, which they then use to evaluate and compare current behavior. When deviations from this baseline are detected, the system can flag these anomalies for further investigation. This approach not only helps identify potential security threats, but also reduces the likelihood of false positives, as the system continually refines its understanding of normal behavior over time.

II.4.2. Intrusion Detection

ML-based intrusion detection systems play a vital role in protecting IoT devices against various types of attacks, including botnets and ransomware [90, 91]. These systems leverage the power of artificial intelligence to create autonomous security mechanisms that minimize human intervention and improve the overall resilience of IoT systems [92]. Using ML algorithms to analyze network traffic and device behavior, these systems can detect unusual patterns that may indicate an intrusion.

Incorporating ML into intrusion detection allows for more sophisticated and accurate identification of potential threats. Unlike traditional intrusion detection systems that rely on predefined rules, ML-based systems can adapt to new and evolving threats by learning from previous incidents. This adaptability is particu-

larly important in the context of the Internet of Things (IoT), where the diversity of devices and the complexity of networks present unique challenges. By continuously learning and evolving, ML-based intrusion detection systems can provide a robust defense against a wide range of cyber threats, ensuring the security and integrity of IoT networks.

II.4.3. Predictive Analytics and Malware Detection

Furthermore, machine learning techniques offer a proactive approach to IoT security by predicting and preventing cyber-attacks before they occur [93]. By implementing security mechanisms based on machine learning models such as the Ridge Classifier, Firefly Algorithm, and artificial neural networks, researchers aim to establish robust defense mechanisms for IoT networks [93, 94, 95]. These models improve the detection capabilities of IoT security systems, and contribute to the establishment of secure communication channels within IoT ecosystems [96].

Predictive analytics in IoT security involves analyzing historical data to forecast potential security incidents. ML models can identify patterns and trends that precede cyber-attacks, allowing organizations to take preemptive measures to mitigate risks. Malware detection, another critical application of ML, involves identifying malicious software that may compromise IoT devices. By analyzing the behavior of applications and network traffic, ML algorithms can detect and isolate malware before it causes significant damage. The integration of machine learning with cryptographic techniques like elliptic curve cryptography further strengthens the security posture of IoT infrastructures [83].

II.4.4. Cloud-Assisted Security Services

Researchers have also explored the role of ML in securing IoT systems through cloud-assisted security services [97]. By leveraging ML for intrusion detection and threat intelligence, IoT devices can benefit from cloud-based security solutions that enhance their resilience against cyber threats [97]. Cloud-assisted security services provide a scalable and efficient way to implement advanced security measures across diverse IoT environments.

The integration of cloud computing with machine learning offers several advantages for IoT security. Cloud-based platforms can process and analyze large volumes of data generated by IoT devices, providing real-time insights into potential security threats. This centralized approach allows for the deployment of sophisticated ML algorithms that may be too resource-intensive to run on individual IoT devices. Additionally, cloud-assisted security services can facilitate the

sharing of threat intelligence across different IoT deployments, enabling a collaborative approach to cyber security. By taking advantage of the power of the cloud, organizations can enhance the security of their IoT ecosystems while maintaining flexibility and scalability.

II.5. Cyber Deception for IoT Security

Table II – *ML Techniques for IoT Security & Honeypots*

ML Technique	Application in IoT Security	References
Supervised Learning	Used for malware classification and anomaly detection.	[98]
Unsupervised Learning	Clusters network behavior to detect anomalies and unknown threats.	[99]
Reinforcement Learning (RL)	Optimizes cyber deception strategies and attacker engagement.	[76]
Deep Learning (DL)	Enhances intrusion detection and network behavior analysis.	[100]
Generative Adversarial Networks (GANs)	Simulates realistic attack scenarios to train security models.	[101]

Researchers have delved into various aspects of cyber deception to enhance IoT security. Studies have explored the optimal deployment of defense mechanisms, such as honeypots, within IoT networks through game-theoretic models to analyze and bolster security [71]. Additionally, the development of cyber deception frameworks tailored for specific IoT environments, like 5G networks, has been investigated to address resource constraints, low latency demands, and security challenges inherent in IoT and cellular applications [100]. Table II summarises the ML techniques for IoT security and honeypots.

Strategic Use of Deception

The strategic use of deception, like deploying honeypots, has gained traction among administrators to improve the security of IoT networks by misleading and deterring attackers [102]. Cyber deception strategies aim to inject false information into the attackers' reconnaissance and planning phases, rendering their subsequent actions ineffective or harmless [78]. By predicting likely attack paths and strategically placing decoy nodes along these paths, cyber deception can disrupt and mislead adversaries, ultimately strengthening IoT security [76].

Cyber Ranges and Training

In cyber deception, the concept of a cyber range has been proposed as a training ground to enhance knowledge and awareness in IoT security [72]. The purpose of this approach is to bridge the skills gap in IoT cybersecurity by providing a platform for research, training, and experimentation in IoT security protocols and defense mechanisms [101]. Furthermore, the use of artificial intelligence and expert systems in cyber security has been highlighted as crucial for safeguarding IoT systems, given their susceptibility to breaches due to the interconnected nature of IoT devices [77].

Emerging Trends and Future Directions

As the IoT landscape continues to expand rapidly, the security concerns surrounding IoT devices have become more pronounced. Inadequate security measures coupled with the proliferation of IoT devices pose significant cyber security risks, necessitating robust security frameworks and mechanisms to safeguard IoT ecosystems from a myriad of cyber threats [99]. The integration of architectural-based security mechanisms, encryption techniques, secure protocols, and authentication protocols like password protection has been proposed to mitigate security vulnerabilities in IoT architectures [98].

Cyber deception strategies and technologies play a pivotal role in fortifying IoT security against sophisticated cyber threats. By leveraging advanced techniques such as AI, blockchain, and machine learning, alongside the strategic deployment of deception mechanisms like honeypots, IoT networks can enhance their resilience and thwart malicious activities. As the IoT landscape continues to evolve, it is imperative for researchers and practitioners to stay abreast of emerging trends in cyber deception to effectively safeguard IoT ecosystems from a diverse range of cyber threats.

II.5.1. IoT Honeypots in Cybersecurity

Deception can be classified into six distinct categories: perturbation, obfuscation, moving target defense, mixing, honey-x, and attacker engagement [103]. A honeypot, which is a type of honey-x, is primarily employed to mislead attackers away from their real target or to gather data about attack patterns. Honeypot is a technology designed to capture attacks on IoT devices by diverting attackers away from critical systems and collecting data on their tactics, techniques, and

procedures (TTPs). There are primarily two types of honeypots: *high* and *low* interaction. A hybrid or medium interaction type exists between these two. Beyond these categories, we propose a new level, intelligent-interaction honeypots, which utilize ML. [8] reviewed all honeypot research projects conducted since 2002.

Honeypots are decoy systems designed to attract attackers and gather information about their activities. They serve as a valuable tool in cyber deception by diverting attackers away from critical systems and collecting data on their tactics, techniques, and procedures (TTPs).

II.5.1.1. Low-Interaction Honeypots

Low-interaction honeypots are widely used in cybersecurity due to their simplicity and low resource requirements. These honeypots emulate certain services and protocols to attract and interact with attackers, albeit at a minimal level. One of the most well-known low-interaction honeypots is Honeyd, which can simulate multiple network services and devices, making it a versatile tool in cybersecurity research [104]. The primary advantage of low-interaction honeypots lies in their ability to gather data on common attack methods and scanning techniques without requiring extensive resources or risking significant system compromise.

Low-interaction honeypots have been employed effectively in IoT environments. For example, U-PoT and ThingPot are designed to emulate specific IoT devices, focusing on capturing interactions with those devices [105, 106]. U-PoT simulates universal plug and play (UPnP) services to attract attackers, while ThingPot emulates various IoT devices to study attack patterns and behaviors. Despite their benefits, these honeypots have limitations. Their fixed behavior and limited interaction capabilities make them easily detectable by sophisticated attackers. Moreover, since they only emulate specific functions, they may not provide comprehensive insights into more complex and nuanced attack strategies.

II.5.1.2. High-Interaction Honeypots

High-interaction honeypots offer a more in-depth approach to cybersecurity by providing real systems for attackers to interact with, thereby collecting detailed information on advanced attack techniques. These honeypots mimic full-fledged operating systems and applications, allowing attackers to fully engage with the system as they would with an actual target. The SIPHON architecture is a notable example of a high-interaction honeypot designed for IoT security, deploying physical devices in a controlled environment to gather extensive data on attack methodologies [107].

Another high-interaction honeypot, Honware, exposes firmware images in a virtual environment to attract and study attackers [108]. This approach enables researchers to observe sophisticated attacks, such as those exploiting firmware vulnerabilities, in a realistic setting. However, the complexity and maintenance requirements of high-interaction honeypots pose significant challenges. Deploying and maintaining these systems is resource-intensive, and as the number of honeypots increases, scalability becomes a concern. Additionally, high-interaction honeypots can become targets for exploitation themselves, as seen in instances where honeypots were compromised during DDoS attacks.

II.5.1.3. Medium-Interaction Honeypots

Medium-interaction honeypots aim to balance the advantages of both low-interaction and high-interaction honeypots by offering more interaction capabilities than the former while being less resource-intensive than the latter. These honeypots often use hybrid honeynet systems that combine emulated services with real or virtualized operating systems. IoTPOT is an example of a medium-interaction honeypot that simulates Telnet services for various IoT devices, capturing detailed information on Telnet-based intrusions [109].

The hybrid approach of medium-interaction honeypots allows them to provide more realistic interaction scenarios without the extensive resource requirements of high-interaction honeypots. This makes them suitable for large-scale deployments where resource constraints are a concern. However, they still face challenges related to scalability and the complexity of accurately simulating diverse IoT devices and protocols. Despite these challenges, medium-interaction honeypots represent a promising direction for enhancing IoT security by providing a more comprehensive and resource-efficient method for studying cyber threats.

II.5.1.4. Intelligent-Interaction Honeypots

Intelligent-interaction honeypots focus on engaging with attackers to increase the chances of capturing attacks, rather than precisely replicating the behavior of a specific service or device, as seen in high, low, or hybrid interaction honeypots. However, a few research models stand out for their versatility, as they emulate complete devices and are self-adaptive: IoTCandyJar [73], Chameleon [74], and FirmPot [75].

IoTCandyJar [73], introduced by Luo et al., is an intelligent-interaction IoT honeypot capable of emulating various IoT devices. For any given request, IoTCandyJar selects the response that the attacker is likely expecting, using a collection of responses from IoT devices gathered through internet scanning. If the

chosen response matches the attacker's expectations, they believe the honeypot is their target device and proceed to send exploit code. The system utilizes a Markov Decision Process (MDP) model to learn from scratch what responses attackers anticipate. However, the honeypot takes time to learn how to respond correctly, with the model requiring two weeks to learn enough to maintain continuous interaction with attackers. Additionally, a limitation is that the responses collected through internet scanning may not always originate from actual IoT devices, as some may come from other honeypots on the internet.

Chameleon [74], introduced by Zhu et al., is a self-adaptive IoT honeypot capable of emulating various IoT devices. Chameleon consists of three main components: the front-end responder, evaluator, and back-end interactor. The front-end responder handles incoming requests and provides appropriate responses. For unfamiliar requests, the responder forwards them to the evaluator, which assesses the request's security using an IP whitelist. If the source is deemed untrusted, Chameleon replies with a default response, and the request is saved for further manual analysis. The back-end interactor connects to the targeted IoT device, identifies open ports and services, and activates them on Chameleon. As it processes more requests, Chameleon gradually adopts characteristics similar to the target IoT devices.

FirmPot [75], developed by Liu et al., is a framework that automatically creates intelligent-interaction honeypots using firmware. This system gathers web interactions by emulating firmware within a Docker container and uses RL to understand the relationship between requests and responses. The generated honeypots then provide the most appropriate emulated response to each incoming request. While FirmPot introduces some level of dynamic in its deception strategy, some challenges remain.

- First, the honeypots produced by this framework are unable to detect advanced attacks, such as configuration changes. This limitation is due to several factors: (i) the location and timing of honeypot observation may be inadequate; (ii) the Seq2Seq learning model used by the authors may fail to converge or interact properly; or (iii) in the worst case, attackers may have detected the honeypots, a common issue for many existing honeypots.
- Second, during the scanning process, some responses collected might originate from fake web applications, similar to other honeypots.

- Lastly, this approach heavily relies on firmware images, meaning vulnerabilities in the target vendor's firmware could be exploited when generating the honeypot.

II.5.2. Fake Vulnerability for IoT Device

The concept of deploying fake vulnerabilities in IoT devices has emerged as a strategic approach to mislead attackers and gather intelligence on their tactics and techniques. By intentionally introducing decoy vulnerabilities, security researchers can create a controlled environment to study attack patterns and behaviors, which can be used to improve the overall security posture of IoT systems.

According to research by Thompson et al., fake vulnerabilities can be used to divert attackers away from critical systems and towards honeypots that simulate real-world IoT devices [110]. This method not only protects the actual devices but also provides a useful understanding into the attack vectors and methodologies employed by malicious actors. However, the deployment of fake vulnerabilities must be carefully managed to avoid unintended consequences. Misconfiguration or inadequate isolation of these vulnerabilities could potentially expose real systems to risk. Additionally, attackers who detect the ruse may retaliate with more sophisticated and targeted attacks, as noted by Kim et al. (2021) [111].

Overall, while the strategy of using fake vulnerabilities presents promising benefits for IoT security, it necessitates a meticulous and well-planned implementation to ensure its effectiveness and safety.

II.5.3. Honeypot for IoT Camera

Honeypots have been widely utilized in cyber security to detect, deflect, and analyze attempts at unauthorized use of information systems. When applied to IoT cameras, honeypots can play a critical role in understanding the unique security challenges and threat vectors associated with these devices.

IoT camera honeypots are designed to mimic the behavior and functionalities of real cameras, enticing attackers to interact with them. This interaction allows security researchers to collect data on the attackers' methods and intentions. One notable implementation is the SIPHON architecture proposed by Guarnizo et al. [112], which utilizes high-interaction IoT honeypots hosted in cloud environments to monitor and analyze sophisticated attack patterns.

Ziaie et al. [113] introduced a low-interaction IoT camera honeypot that leverages recorded video content to simulate real-time streaming. This approach provides a less resource-intensive solution while still capturing valuable data on intru-

sion attempts . Another innovative model is IoTCMal, which employs authentic IoT cameras to create an engaging and believable honeypot environment for attackers [5].

HoneyCam, developed by Guan et al. [6], replicates the functionalities of IoT camera video streaming services using pre-recorded 360° video content. While effective in attracting attackers, its static nature limits real-time interactivity, which is crucial for detecting more advanced and dynamic attack strategies.

Despite the effectiveness of IoT camera honeypots, they face several challenges. High-interaction honeypots, like SIPHON, require significant resources for deployment and maintenance. Furthermore, the static nature of low-interaction honeypots, such as HoneyCam, may not provide comprehensive insights into more complex attack vectors. Researchers suggest combining different types of honeypots to create hybrid systems that balance resource efficiency and interaction depth, enhancing the overall capability to detect and analyze cyber threats.

Despite significant advancements in leveraging machine learning and cyber deception for IoT security, several gaps and challenges remain. The current literature predominantly focuses on static and predefined deception strategies, which, although effective, may not be sufficient to combat the rapidly evolving threat landscape of IoT ecosystems. While machine learning techniques have been extensively used for anomaly and intrusion detection, their integration with dynamic and adaptive deception strategies remains underexplored. Additionally, existing approaches often lack the ability to adapt in real-time to new and unforeseen attack vectors, leading to potential vulnerabilities. The deployment of reinforcement learning in IoT security is emerging but still in its nascent stages, with limited practical implementations. This thesis aims to address these gaps by introducing an adaptive cyber deception framework that combines reinforcement learning and machine learning techniques. By dynamically adjusting deception mechanisms based on real-time threat intelligence and environmental feedback, this approach seeks to enhance the resilience of IoT systems against sophisticated cyber threats, providing a robust and proactive security solution. In this thesis, we powered the deception strategy by applying cutting-edge technology such as recommender systems and augmented reality combined with reinforcement learning.

II.6. Conclusion

In this chapter, we provided a comprehensive overview of the foundational concepts essential for understanding IoT security, cyber deception, and ML. We began

by exploring the definition, evolution, architecture, technologies, and protocols of the IoT, highlighting the significant security challenges it poses. We then examined the concept of cyber deception, emphasizing its role in enhancing cybersecurity by proactively misleading and confusing attackers. Finally, we introduced machine learning, detailing its types and applications in cybersecurity, particularly for threat detection, anomaly identification, and response automation.

Despite the significant advancements in leveraging ML and cyber deception for IoT security, several gaps and challenges remain in the existing literature. The current research predominantly focuses on static and predefined deception strategies, which, although effective, may not be sufficient to combat the rapidly evolving threat landscape of IoT ecosystems. For instance, many existing honeypot systems, such as low-interaction and high-interaction honeypots, rely on fixed behaviors and predefined responses, making them susceptible to detection by sophisticated attackers. Additionally, while ML techniques have been extensively used for anomaly and intrusion detection, their integration with dynamic and adaptive deception strategies remains underexplored. This limitation restricts the ability of IoT security systems to adapt in real-time to new and unforeseen attack vectors, leaving potential vulnerabilities unaddressed.

Furthermore, the deployment of RL in IoT security is still in its nascent stages, with limited practical implementations. RL has the potential to create autonomous and adaptive security mechanisms that can learn and evolve based on real-time threat intelligence and environmental feedback. However, the current literature lacks a comprehensive framework that effectively combines RL with cyber deception to enhance the resilience of IoT systems against sophisticated cyber threats. Another notable gap is the lack of scalability in existing deception-based security solutions, particularly in large-scale IoT environments where resource constraints and diverse device

III

CHAPTER

Contribution to Adaptive Intelligent Cyber Deception for Probing Attacks on IoT Devices

CONTENTS

III.1 - Introduction 48
III.2 - Probing Attacks on IoT Devices 49
III.3 - Adaptive Intelligent Interaction Honeypot for Internet of Things . . 54
III.4 - Evaluation and Results 65
III.5 - Analysis and Discussions 69
III.6 - Conclusion 73

III.1. Introduction

With the growing number of connected devices, the security of IoT systems has become a major concern due to their various vulnerabilities. Unlike traditional computers, IoT devices usually come equipped with network interfaces that enable interaction between the physical and digital worlds. However, they are often susceptible to multiple vulnerabilities, including the use of weak or hard-coded passwords. Many of these passwords are easy to guess, publicly accessible, or unchangeable, which makes the devices vulnerable to being easily compromised. Probing attacks, which seek to identify and exploit these vulnerabilities, are particularly challenging to defend against. Cyber deception techniques, such as honeypots, have been employed to mislead attackers and protect IoT devices. IoT vulnerabilities are often closely tied to the specific device, its firmware version, or even the manufacturer. As a result, once attackers scan the network and identify

vulnerabilities, such as open ports, they typically conduct further assessments on the targeted device to gather more detailed information before initiating an attack using exploit code.

Our objective is to design a honeypot capable of interacting with attackers during the reconnaissance phase, allowing us to effectively monitor attacks targeting IoT devices and gain access to the exploit code used by the attacker. To accomplish this, the honeypots need to successfully pass the attacker's initial pre-check phase. Given the vast number and diversity of IoT devices, this poses a significant challenge, traditional security techniques are not affordable. Even though, Some device checks are straightforward, like confirming that the target device provides the expected response. A honeypot that delivers the correct response to a request may be able to bypass these checks. However, other checks are more complex and involve additional steps before the exploit code is executed.

This chapter introduces an advanced approach to cyber deception using intelligent adaptive interaction honeypots to counter probing attacks on IoT devices. We develop and evaluate an adaptive intelligent interaction honeypot named AIIPot, designed to enhance the security of IoT devices. It combines ML techniques, including BERT and GAN models, to create a sophisticated deception mechanism that interacts with attackers, gathers valuable attack data, and improves the overall security posture of IoT networks. By utilizing RL, adaptive honeypots can learn from interactions and improve their deception capabilities over time, making them more effective against sophisticated attacks.

This chapter is structured as follows: Section III.2 provides a detailed overview of probing attacks, their phases, common techniques, and the role of honeypots in cybersecurity. Futhermore, this section gives some background on BERT, GAN and RL. Section III.3 introduces our contribution, AIIPot, detailing its architecture, components, and the integration of ML models to enhance its effectiveness. In addition, this section presents the setup, data collection, and analysis of experimental results to demonstrate the performance of AIIPot in real world scenarios. Finally, Section III.6 summarizes the key findings, discusses the implications for IoT security, and suggests directions for future research.

III.2. Probing Attacks on IoT Devices

Probing attacks are a type of cyber attack in which attackers systematically scan a network to gather information about its configuration and identify potential vulnerabilities. These attacks are typically the first step in a more extensive

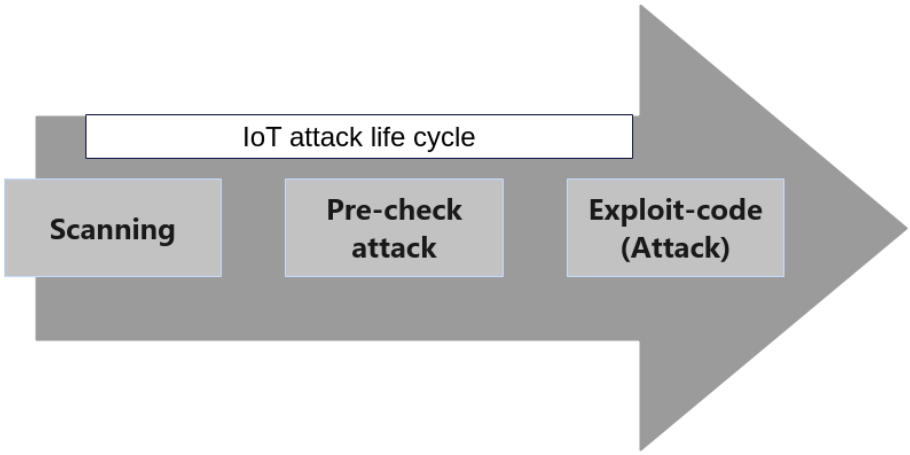


Figure 6 – *Life cycle of IoT attack.*

campaign aimed at exploiting the identified weaknesses. Probing attacks can be divided into several phases: the reconnaissance phase, where attackers collect as much information as possible about the target network and devices, including identifying active devices, open ports, and running services; the scanning phase, where attackers perform more focused scans to map out the network and pinpoint specific vulnerabilities; and the enumeration phase, where attackers gather detailed information about the network's structure, user accounts, and system configurations to prepare for the actual attack.

III.2.1. Phases of Probing Attacks

Probing attacks generally follow a structured lifecycle, each phase designed to extract different types of information about the target network. Understanding these phases is crucial for developing effective defensive measures.

III.2.1.1. Reconnaissance Phase

In the reconnaissance phase, attackers seek to gather preliminary information about the target network. This phase involves passive techniques such as monitoring network traffic and actively probing the network using tools like IPsweep. IPsweep sends packets to a range of IP addresses to identify which ones are active, helping attackers map out the network. This phase is crucial as it helps attackers identify potential targets without alerting the network defenders.

Additionally, reconnaissance provides a foundation for subsequent attack phases by revealing key details about the network's topology and active devices. Attackers often use this information to craft more precise and effective scans, minimizing their chances of detection. The effectiveness of the reconnaissance phase often determines the success of the entire attack campaign.

III.2.1.2. Scanning Phase

The scanning phase involves more direct and detailed probing of the network. Tools like Nmap [114] are commonly used in this phase. Nmap [114] provides detailed information about network hosts, open ports, and running services, and can perform various types of scans, including TCP SYN scans, UDP scans, and OS fingerprinting. This phase allows attackers to identify specific vulnerabilities that can be exploited in the later stages of the attack.

Moreover, the scanning phase often includes vulnerability scanning, where attackers use specialized tools to detect known vulnerabilities in network devices and software. This phase helps attackers to rank their targets according to the seriousness of the detected vulnerabilities, increasing the efficiency and impact of their attacks. Scanning results are crucial for attackers to develop a comprehensive map of the network's weaknesses.

III.2.1.3. Enumeration Phase

In the enumeration phase, attackers gather detailed information about the network's structure, user accounts, and system configurations. This phase involves more intrusive techniques, such as querying network services and extracting data from network devices. The goal is to gain as much detailed information as possible to prepare for the actual attack, ensuring that the exploit phase is effective and precise.

Enumeration is critical as it often involves identifying valuable assets within the network, such as databases, user credentials, and sensitive information. Attackers use this information to refine their attack strategies and develop specific exploits tailored to the target environment. Effective enumeration can significantly increase the success rate of the subsequent attack phases.

III.2.2. Techniques Used in Probing Attacks

Different techniques are employed during probing attacks to gather comprehensive information about the target network. Each technique serves a specific purpose and helps attackers build a detailed map of the network's vulnerabilities.

III.2.2.1. IPsweep

IPsweep [115] involves sending packets to a range of IP addresses to determine which addresses are active. This technique helps attackers identify live hosts on a network, providing a foundation for more detailed scanning. By mapping out the active devices, attackers can focus their efforts on potential targets, reducing the scope of their attack and minimizing the risk of detection.

Additionally, IPsweep can be used to identify the network's perimeter devices, such as routers and firewalls, which are critical for understanding the network's defensive posture. This information helps attackers plan their intrusion methods, such as choosing the best entry points and avoiding detection by security systems.

III.2.2.2. Nmap Scans

Nmap [114] is a flexible network scanning tool that offers comprehensive details about network hosts, available ports, and active services. It supports various types of scans, including TCP SYN scans, UDP scans, and OS fingerprinting. Nmap helps attackers identify specific vulnerabilities in the network, such as open ports and unpatched services, that can be exploited in subsequent phases of the attack.

Furthermore, Nmap's ability to perform OS fingerprinting allows attackers to determine the operating systems running on target devices, enabling them to tailor their exploits to the specific software versions. This detailed information significantly enhances the precision and effectiveness of the attack, making Nmap an essential tool in the attackers' arsenal.

III.2.2.3. Portsweep

Portsweep [115] focuses on scanning a single host for open ports. This technique allows attackers to identify services running on a specific device, which can then be targeted for exploitation. By determining which ports are open and what services are running, attackers can tailor their attacks to exploit specific vulnerabilities associated with those services.

Portsweep is particularly effective for identifying high-value targets within the network, such as servers running critical applications. Once identified, these targets can be subjected to more intensive probing and exploitation, potentially yielding significant rewards for the attackers.

III.2.3. Impact of Probing Attacks on IoT Security

Probing attacks pose significant risks to IoT security due to the inherent vulnerabilities of IoT devices. These devices usually lack powerful security measures and are more exposed to probing attacks, making them attractive targets for cyber attackers.

III.2.3.1. Exploitation of Vulnerabilities

Once vulnerabilities are identified through probing, attackers can use them to acquire unauthorized access, disrupt services, or steal sensitive data. For example,

exploiting an open port on an IoT device can allow attackers to bypass authentication mechanisms and gain control over the device. This can lead to various malicious activities, such as eavesdropping on communications, manipulating device operations, or launching further attacks on the network.

The exploitation of these vulnerabilities can have severe consequences, especially in critical infrastructure or healthcare environments where IoT devices play essential roles. The absence of security measures in these devices can lead to significant disruptions and compromise the protection and privacy of users.

III.2.3.2. Preparation for Sophisticated Attacks

Probing attacks often serve as a precursor to more sophisticated attacks, such as denial-of-service (DoS) attacks or the deployment of malware. By gathering detailed information about the target network during the probing phase, attackers can plan and execute more effective attacks. For instance, knowledge of the network's topology and device configurations can help attackers identify critical points of failure and target them to maximize the impact of their attack.

In Addition, the information collected during probing can be utilised to craft targeted phishing attacks or social engineering tactics, further compromising the security of the network. This multi-phase approach allows attackers to systematically weaken the network's defenses and increase their chances of success.

III.2.3.3. IoT Device Vulnerabilities

IoT devices are particularly vulnerable to probing attacks due to their diverse nature and often limited security measures. The firmware version, device configuration, and even the vendor can introduce specific vulnerabilities. Attackers, after scanning the network and identifying potential targets, perform additional checks on the remote target devices to gather more detailed information before launching their exploit code. The pre-check phase is essential for guaranteeing the effectiveness of the attack. This pre-check phase is essential for guaranteeing the effectiveness of the attack.

IoT devices often operate in environments where security updates are infrequent, and default settings remain unchanged, making them easy targets for attackers. The diversity of IoT devices also means that a single vulnerability can lead to widespread exploitation across multiple devices and networks.

III.2.3.4. Pre-check Attack Step

During the pre-check attack step, attackers verify the presence of specific vulnerabilities by interacting with the target device. This step helps attackers confirm

that the device is vulnerable to their planned exploit. If a honeypot or security mechanism fails to convincingly emulate the target device's behavior during this step, attackers may realize they are being deceived and avoid triggering the exploit, thus evading capture.

This pre-check step underscores the importance of developing sophisticated honeypots that can accurately mimic real devices and engage attackers convincingly. Effective pre-check detection and response mechanisms are essential for preventing attackers from progressing to the exploitation phase.

Figure 6 illustrates the stages of an IoT attack. A honeypot with a low level of interaction typically cannot pass the pre-check phase, resulting in its inability to capture the actual attack. Therefore, developing more sophisticated honeypots that can accurately emulate target devices and engage attackers convincingly is crucial for effective IoT security.

Even though researchers have proposed intelligent interaction honeypots that add the likelihood of catching the attack, as presented in Chapter II, these methods have significant limitations. They often lack convergence, leaving open doors in the network and failing to adapt to new vulnerabilities and attackers over extended periods. These weaknesses reduce their effectiveness in providing long-term engagement with attackers and maintaining robust security. Consequently, the existing solutions fall short in creating a truly adaptive and resilient defense mechanism capable of responding to the rapidly evolving threat landscape in IoT environments.

To address these challenges, we propose an advanced honeypot for IoT devices that utilizes ML principles to effectively respond to attackers during the early stages of an attack. Our solution employs sophisticated machine learning techniques to autonomously learn from and engage with attackers. By incorporating RL, the system adapts in real-time, improving its ability to capture exploit code and providing strong defense against complex attacks. This approach is designed to enhance the flexibility and durability of honeypots, ensuring they stay effective against evolving cyber threats.

III.3. Adaptive Intelligent Interaction Honeypot for Internet of Things

In this section, we provide a detailed explanation of our method for developing a honeypot specifically for IoT devices.

III.3.1. System Overview

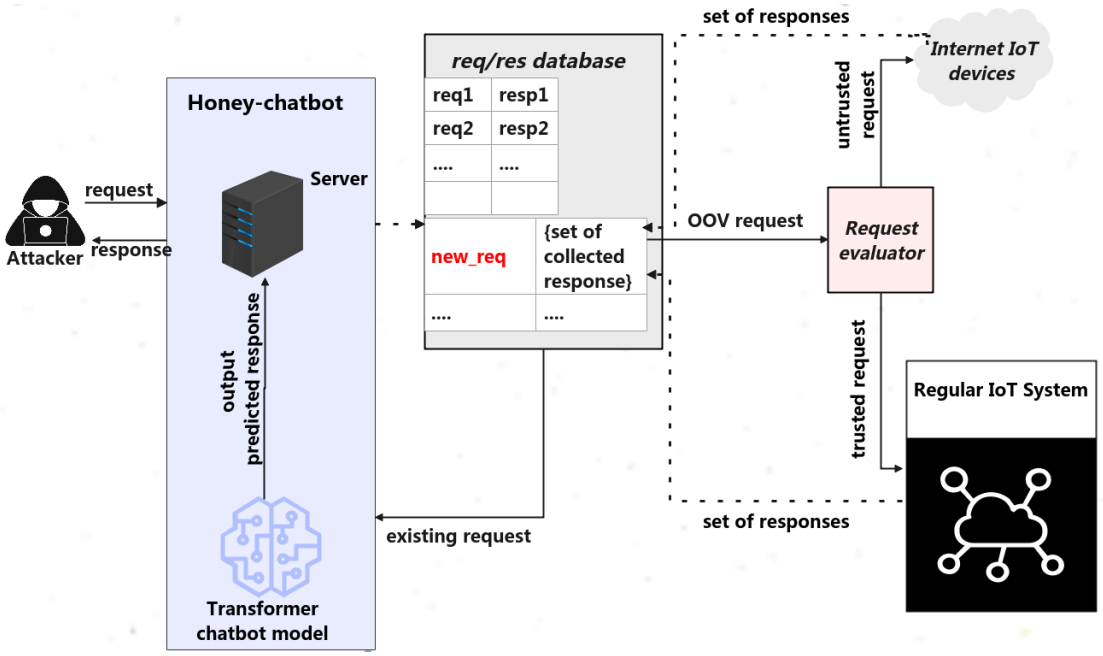


Figure 7 – *High-level overview of AIIPot.*

Figure 7 presents a high-level summary of our approach. The framework consists of three main components: the *honey-chatbot*, the *req/res database*, and the *request evaluator*. This system assumes a scenario where the attacker is already engaging with the honeypot, which is why the architecture depicts the attacker as the sole entity interacting with the system. The *honey-chatbot* is responsible for responding to attacker requests within the honeypot, functioning as a server where the response selection model is deployed. This component handles interactions with attackers by providing replies.

To initially train the ML model, we collected a dataset following the method suggested by FirmPot [75]. This dataset is stored in the *req/res database*, which contains possible requests that attackers might send to an IoT device along with the corresponding responses from the IoT device. If a request is not found in the *req/res database*, it is treated as an *out-of-vocabulary* (OOV) request. The OOV request is then recorded as a new entry in the *req/res database* and sent to the *evaluator* module for security assessment. The evaluator determines the trustworthiness of the request and, if it is considered safe[1], it is broadcast to all devices within the local network. Otherwise, untrusted requests are forwarded to the IoT device on the internet.

1. The request does not contain any exploit code.

III.3.2. Req/Res Database and Request Evaluator

This section provides a detailed explanation of how we compile the req/res database component in our proposed approach.

The Honey-Chatbot component utilizes this database during the offline training phase of the transformer model[2].

We used the dataset from [75] as the foundation for our database. Each entry in the database corresponds to a specific request made by an attacker, along with the IoT device's response. For new requests, the *Request Evaluator* is employed to assess trusted requests before capturing the associated response. All new request/response pairs are added to the database, as illustrated in Figure 8.

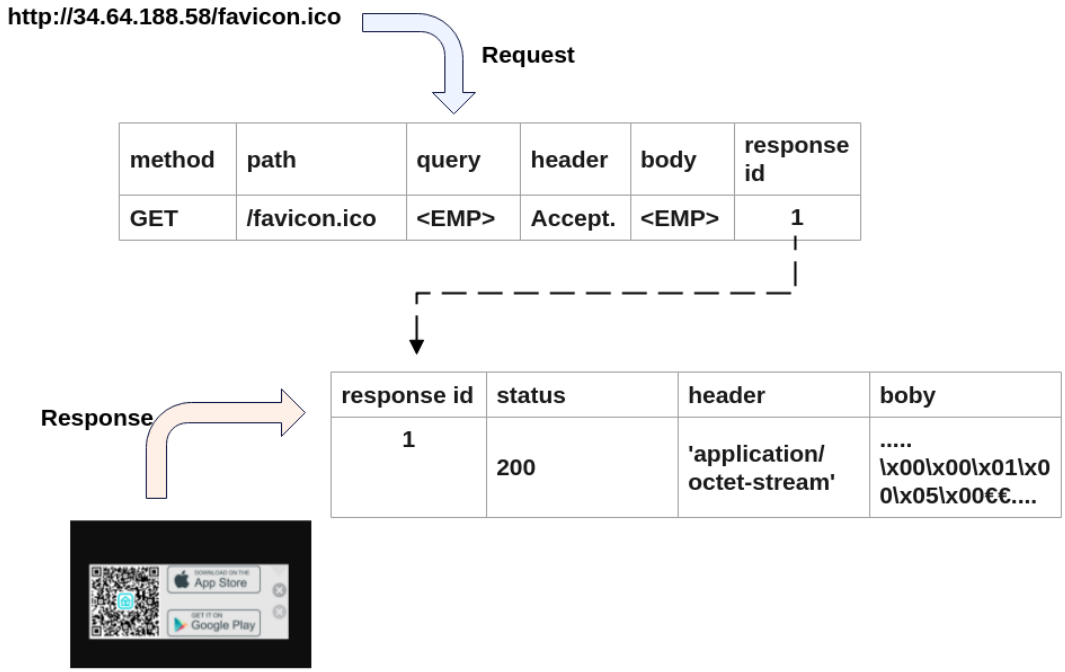


Figure 8 – *Req/res database entries.*

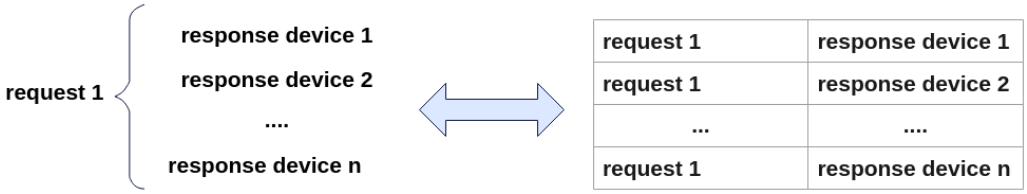


Figure 9 – *Request/response multi-entries.*

2. In an offline machine learning model, the weights and parameters are updated as the model learns, aiming to minimize the global cost function based on the training data.

III.3.3. Intelligent Interaction Honeypot: Honey-Chatbot

The Honey-Chatbot module functions as a server where the response model selection is implemented. Utilizing the req/res database, the honeypot can provide a valid response to the client based on the request received, rather than giving a fixed response. In this section, we explain how to use ML and RL models to optimize response selection, maximizing the chances of capturing attacks.

For each request, the req/res database may contain hundreds or even thousands of possible responses, as shown in Figure 9. While all of these responses are valid, only a few are the correct or expected ones. This is because different IoT devices process the same request in their own way, producing various responses. A simple example is a request to access the root path of a web service: some devices might return a login page, others might redirect to a different resource, and some may show different types of error pages. As a result, all responses in the req/res database are potential options for the attacker's request, but the challenge lies in identifying the one that matches the attacker's expectations.

III.3.3.1. BERT-based Honeypot

Our approach involves initially training a transformer-based architecture (BERT) model on the dataset stored in the req/res database, identifying potential response candidates that the attacker is likely expecting. We then use an MDP model to select the most probable expected response and record the attacker's next action. We assume that by choosing the correct response, attackers will perceive the honeypot as the vulnerable IoT device they are targeting, thus extending the session and eventually delivering the malicious payload. Each incoming request to the honeypot is processed by this module, and the selected response is sent back to the client.

We began by creating a dictionary that assigns numerical values to the words in each request. Then, a word embedding is trained from the numerical vectors representing the requests and responses, denoted as x_t and y_t, respectively. This trained word embedding translates similar words into similar vectors. The vectorized requests and responses are then input into a transformer model. We extend BERT by adding task-specific layers, similar to the standard BERT fine-tuning process. The encoder's output is passed through a linear layer followed by a softmax layer, which produces $\pi_\theta(y_t|x_t)$, the probability of selecting the response y_t for a given request x_t.

Ideally, the output with the high probability value is consider as the response to sent to the attacker at that stage. This response is drawn from a real dataset con-

taining the real information about IoT device are response from real IoT devices, this meant an intelligent and sophisticated attacker can used those to prepare more sophisticate attacker in the future. So, the BERT model will then be see at this case as an additional vulnerability to our network. In other to solve this problem, we propose a GAN & BERT model that still provide intelligent and blur response to attacker. The response sent by this model are likely similar to reals ones but not real.

III.3.3.2. GAN & BERT-based Honeypot

We explore the combination of BERT model with GAN to provide blur response and enhance the convergence speed of the model thereby improving the response selection process, enabling effective capture of attacks.

We consider that the attacker may used a machine to make use of the real responses provided by BERT, which is a typical case of the GAN. This is because the attacker and defender are both using an ML model. In a typical situation of defensive deception [103], the attacker is considered the discriminator and the defender the generator. The goal of the generator is to fool the discriminator, as the goal of the defender is to fool the attacker. But the discriminator always tries to discriminate between real and fake information. That is, the attacker wants to discriminate between expected to unexpected information in order to believe the system. A deception system should not only generate expected information but also ensure that it does not reveal enough information to the attacker to gain knowledge about the regular IoT system. This will help the defender to better deceive the attacker and keep the actual system hidden from their view.

Model formulation To formulate the response selection problem with the BERT in an adversarial learning setting. We make two mains assumptions:

(i) Our first assumption is that we are in a situation where the attacker uses an ML model to perform an attack on our system. This is a valid assumption based on the increasing rate of ML models and the intelligence and rationality of attackers.

(ii) The second assumption is based on the idea that the attacker's decision to either continue the session or move forward with the attack is entirely dependent on the response they received to their previous request. This assumption is reasonable given our current understanding of existing malware samples.

Adversarial learning setting : Our approach involves initially fine-tuning a BERT model in an offline environment to learn a vector representation of the dataset contained within the req/res database. In addition to this, train a generative model from a random distribution that learns to generate a response similar to the one on the real database. This generator is considered our defender, which fools the attacker with a fake response. The discriminator is considered an attacker. It discriminates between expected and unexpected responses from the generator. Two scenarios are expected:

- If the response produced is unexpected, the discriminator punishes the generator for producing an unexpected response by ending the session;
- If the response produced is expected, the discriminator rewards the generator for producing an expected response by sending a new request.

  A good discriminator should not reveal enough information for the generator to make progress. Same as an attacker with the goal of hiding his identity do not reveal enough information for the defender to make progress in terms of the response produced.

Since the generator starts with a random noise to produce a response, the convergence will be slow. Also, the generator is trained in an online setting, so the model will need too much time (or punishment from the attacker) to generate an appropriate response. To solve this problem, we consider the fusion of the generator output and the BERT model trained offline.

BERT and adversarial learning model: We begin with a pre-trained BERT model and fine-tune it to learn the representation of our specific req/res dataset. For simplicity, let's consider a sentence classification task with k response categories. Given an input request $x_t = (x_{t_1}, x_{t_2}, ..., x_{t_n})$, BERT outputs vector representations in $\mathbb{R}^d$ as $(h_{CLS}, h_{t_1}, ..., h_{t_n}, h_{SEP})$. We use the h_{CLS} representation as the request embedding for the target tasks.

As illustrated in Figure 10, we enhance the BERT model with a GAN architecture by incorporating (i) a discriminator D that classifies responses as either expected or unexpected, and (ii) a generator G that operates adversarially. Specifically, G is a Multi-Layer Perceptron (MLP) that takes a noise vector from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ as input and outputs a vector representation of the fake response $h_{fake} \in \mathbb{R}^d$.

The discriminator (D) is also a MLP that receives input from a fusion vector $h_{fus} \in \mathbb{R}^d$. The fusion vector h_{fus} is the late fusion of the vector representations

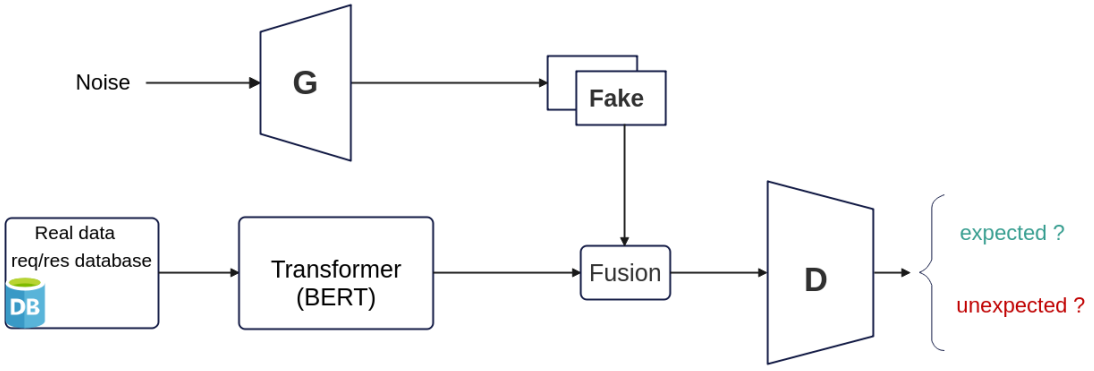


Figure 10 – *BERT with GAN architecture: G generates a set of fake responses F given a random distribution of requests. These, along with the real req/res dataset vector representations computed by BERT, are used as input for the discriminator D.*

produced by the generator and the BERT model. We perform three different types of fusion to obtain the final vector representation of the response output h_{fus}.

- **Fusion 1**: $h_{fus} = \eta h_{fake} + (1-\eta) h_{CLS}$ with $\eta \in [0,1]$
- **Fusion 2**: $h_{fus} = h_{fake} + h_{CLS}$
- **Fusion 3**: $h_{fus} \in_R \{h_{fake}, h_{CLS}\}$

The final layer of the discriminator D is a sigmoid activation layer, which produces an output indicating whether the response is expected or unexpected.

Algorithm: we trained the BERT model offline on a real dataset. As explained below, the idea behind training the BERT model is to make the model learn a good representation of the real dataset. That's learning to output a corresponding response to a given request. The trained BERT model is used in an online setting to ameliorate the performance of the generator and speed up the convergence of the learning model. As shown in Figure 10, for a given request, at the first epoch, the BERT model outputs the vector representation h_{CLS} of the response in the real dataset. The generator G outputs h_{fake}. The two vectors are fused into one h_{fus}. We proposed three types of fusion methods. During the evaluation of the model, only the fusion with the best performance was considered.

- On fusion 1 we start the training with the output produced by the BERT and progressively move to the output produced by the generator. With this technique, the discriminator punishes the generator with a high-loss function every time the BERT model outputs an unexpected response until the generator is able to output an appropriate response. This fusion helps the model to converge faster.

- In the case of fusion 2, h_{fus} is obtained by making a simple concatenation between h_{CLS} and h_{fake}.
- Finally, we propose the case of fusion 3, where at each epoch step, h_{fus} is randomly chosen between h_{CLS} and h_{fake}.

The GAN & BERT model, $\pi_\theta(y_t|x_t)$, generates a probability vector indicating the likelihood of each response y_t being the correct one. However, selecting the response with the highest probability from $\pi_\theta(y_t|x_t)$ does not guarantee that the attacker will send another request. To address this, we employ an MDP to model the ongoing interaction, encouraging the attacker to continue sending requests until the exploit code is eventually transmitted. The key component of the system is the selection engine, which inputs the request into the GAN & BERT model to retrieve a list of possible responses. These responses are then assessed by the MDP model. In MDP selection mode, the system identifies the current state from the normalized request and uses the model to choose the most effective response, aiming to keep the attacker engaged in the interaction.

III.3.4. Adaptive Intelligent Interaction Honeypot

III.3.4.1. Model Formulation

Here, we explain how we frame the response selection problem using the GAN & BERT model in conjunction with the MDP model. We assume that the attacker's decision to either continue the session or launch an attack is influenced by the response to their previous request. This assumption is reasonable based on our current understanding of existing malware samples.

III.3.4.2. Adaptive Mechanisms

Markov Decision Process (MDP): MDP extends the traditional (unhidden) Markov model and is used for sequential decision-making in uncertain environments. It involves computing a policy of actions that maximize utility in relation to expected rewards. At each decision point, the next state is determined by the action taken, based on a transition probability function. This approach is part of RL, which is a method for controlling and adjusting the policy when the rewards associated with the current state space are uncertain.

An MDP, also referred to as a stochastic control problem, is an extension of the traditional (unhidden) Markov model. It serves as a framework for sequential decision-making in situations where outcomes are uncertain, such as determining a policy of actions that maximizes utility based on expected rewards. In each state,

a set of possible actions can be taken, which then transition the system into a new state. At each decision point, the next state is determined by the selected action through a transition probability function. Essentially, it operates like a Markov chain, where state transitions depend entirely on the transition function and the action taken in the prior step. The outcomes of actions (i.e., rewards) and the effects of policies are not always immediately clear, so a mechanism is required to adjust the policy when the rewards in the current state are uncertain. This mechanism is known as RL.

Problem formulation: In the typical RL framework, an agent (attacker) interacts with its environment (honeypot). This interaction involves the agent observing the environment and, based on the input, selecting an action to take within that environment. Every reinforcement learning model develops a mapping from situations to actions through trial-and-error interactions with a changing environment. The model consists of multiple variables, including decision epochs (t), states (x, s), x to denote the request sent by the agent(attacker), transitions probabilities (T), rewards (r), actions (y) corresponds to the response from the honeypot., value function (V), discount (γ), and estimation error (e). The fundamental principle of a reinforcement learning task is represented by the Bellman equation:

$$V^*(x_t) = r(x_t) + \gamma V^*(x_{t+1}). \tag{III.1}$$

The general update policy is expressed as:

$$\Delta w_t = \max_y \ [r(x_t, y) + \gamma V(x_{t+1})] - V(x_t). \tag{III.2}$$

Our problem is essentially a non-deterministic Markov Decision Process, meaning that at each state, there is a transition probability function T that determines the next state. In other words, our learning policy balances probabilistically between *exploration*—responding with new, unused responses—and *exploitation*—responding with known high-reward responses. Calculating the necessary integrals using standard value iteration is impossible without additional knowledge or adjustments in decision-making. Hence, we employ Q-learning to handle the task of maximizing over a set of integrals. Instead of mapping states to state values, Q-learning maps state/action pairs to values, known as Q-values. Q-learning does not use a value function but relies on the Q-function, where each action in a state has a corresponding Q-value. A Q-value represents the total rewards received when performing the associated action and following the given policy afterward.

Thus, in our use of Q-learning, the equivalent Bellman equation is formalized as:

$$Q(x_t, y_t) = r(x_t, y_t) + \gamma \max_{y_{t+1}} Q(x_{t+1}, y_{t+1}). \tag{III.3}$$

The update rule for direct Q-learning, where α is the learning rate, is formalized as:

$$\Delta w_t = \alpha[(r(x_t, y_t) + \gamma \max_{y_{t+1}} Q(x_{t+1}, y_{t+1}, w_t) - Q(x_t, y_t, w_t) \frac{\partial Q(x_t, y_t, w_t)}{\partial w_t}. \tag{III.4}$$

Reward function: The reward function $r : (x_t, y_t) \longrightarrow r$ assigns a value r to the state and action pair (x_t, y_t). Its purpose is to indicate the preference for each pair and to maximize the final rewards, leading to the optimal policy. In our case, the immediate reward $r(x_t, y_t)$ reflects the progress made during the interaction when choosing response y_t for request x_t and transitioning to the next state x_{t+1}. Since progress can be positive or negative, the reward function may also assign positive or negative values. The general rule for defining the reward is that if the response matches the type of device the attacker expects and the attacker follows up by sending exploit code in the next request, the reward should be significantly positive. Conversely, if the response is not as expected (e.g., indicating a non-vulnerable device version), the attacker may abandon the attack, ending the session and leading to a dead-end state, which results in a negative reward. In essence, we reward responses that move us closer to capturing the final attack packet and penalize those that lead to a terminated session. One approach we use is to assign rewards based on the length of the final session, as we assume that the longer the session, the higher the chance of receiving a malicious payload. A standard session length is 2, meaning after our response, there is at least one more request from the same IP on the same port. If no further interaction occurs, a negative reward is given to that response. Alternatively, rewards could be assigned based on whether any known exploit packets are received.

State and action: In our scenario, the state x_t represents the requests sent by the attacker, along with all similar requests stored in the req/res database. The actions are defined by the responses generated by the GAN & BERT model, where $\pi(y_t|x_t) \geqslant threshold$. We aim to set the threshold at 0.5, which is a reasonable assumption because if the GAN & BERT model predicts a response with a probability of 0.5 or higher, there is a strong likelihood that the response belongs to the set of a vulnerable IoT device's responses.

Transition probabilities: The transition probabilities are defined by the transition function $T(s, y, s')$, where y represents the action taken in the current state

s, and s' refers to the resulting new state. More formally, the transition function $T(s,y,s')$ is described by the equation:

$$P(S_t = s' | S_{t-1} = s, y_t = y) = T(s,y,s'). \tag{III.5}$$

To evaluate the probability of each combination of (s,a,s'), we utilized the trained BERT model to select a response from the candidate set, and we saved the session information to the session table. After running the system for some time, we accumulated numerous sessions and analyzed each one to count how often each combination (s,a,s') occurred, denoted as $C(s,a,s')$. The transition function $T(s,y,s')$ is then defined as follows:

$$T(s,a,s') = \frac{C(s,a,s')}{\sum_{x \in S}(s,a,x)}. \tag{III.6}$$

Online Q-learning algorithm: We utilize the online Q-learning algorithm to choose the most likely response from the set of candidate responses generated by the GAN & BERT model. In our Q-learning process, the learning begins when a request is received at the t_0 decision point. After receiving the request, we pass it through the trained GAN & BERT model to generate a set of potential responses. We apply the ε-greedy policy for selecting actions, where the probabilities produced by the BERT model for each candidate response are treated as the initial transition functions. Using this policy, we either select a random action with probability ε, or choose the action that maximizes the reward with a probability of $1-\varepsilon$ for a given state.

Next, we initiate the Q-learning iteration and update the Q-learning table. When we reinforce a specific state-action pair, $r(x_t, y_t)$, we first backpropagate and update the Q-value table accordingly. This allows us to adjust by eliminating responses that lead to negative rewards and refining the ε value. The iteration continues until the model reaches convergence. In practice, our model operates in real-time and updates continuously, so with the aid of the pre-trained GAN & BERT model, the algorithm is more likely to converge efficiently.

III.3.5. Request evaluator

The request evaluator determines whether a request contains exploit code or not. This module relies on an existing work [116] that uses a Support Vector Machine (SVM) classifier to analyse the nature of a request on a network. We evaluated the model proposed by [116] using the NSL-KDD dataset [117, 118] to identify whether a request is an attack or a normal one. Requests classified as attacks are

labelled as untrusted, while those classified as normal are considered trusted. If a request is deemed trusted, it is forwarded to our IoT local network, as depicted in figure 7. Otherwise, the request is sent to accessible IoT devices on the internet. For each request, we received multiple responses from actual IoT devices and stored these responses as separate entries in the database.

III.4. Evaluation and Results

III.4.1. Evaluation of the Learning Model

Here we evaluate the learning process of the ML model that we deployed on GAN-AIIPot.

Learning process of the different fusion method: Based on the training schedule presented previously, the plot in Figure 11 shows the accuracy scores of the models: when Fusion 1, Fusion 2 or Fusion 3 is considered. We observe that the model train with Fusion 1 performs with higher accuracy than the one from Fusion 2 or 3. Deploying the model with Fusion 1 effectively keeps the attacker deceived for a longer period, as the responses generated by this fusion are more precise and appropriate, given the model's accuracy level.

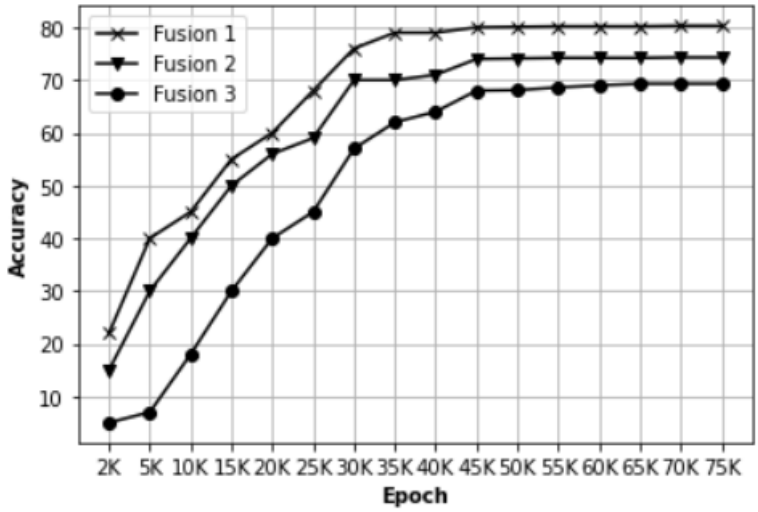


Figure 11 – *Learning curve of the model with different fusion methods.*

Learning process of the GAN & BERT We compare the learning process of the proposed GAN combined with the BERT model with a model that uses only a BERT model. Figure 12 confirms that in terms of accuracy, the GAN & BERT model performs better than the BERT model used in AIIPot.

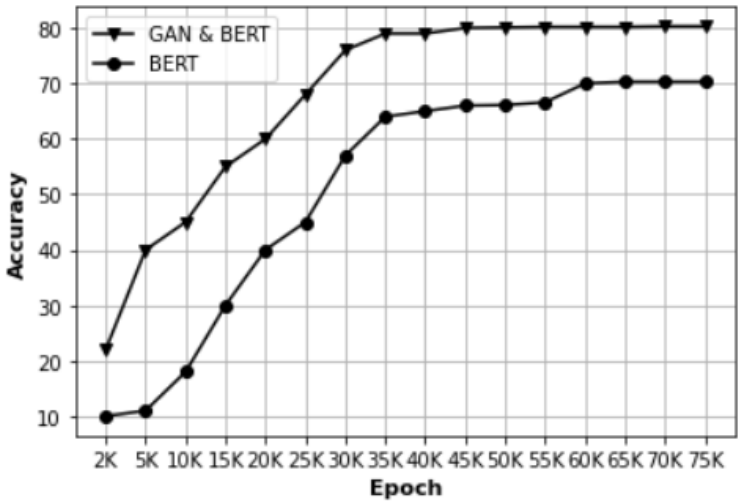


Figure 12 – *Learning curve of the proposed GAN & BERT model with the best fusion method.*

We observe from Figure 12 that the convergence of the combined BERT & GAN model with Fusion 1 is faster.

After training the model with the setup presented in the previous section, we tested the model on unseen data and obtained the result in Figure 13.

Request	http://34.64.188.58/favicon.ico	http://34.64.188.58/webpages/locale/language.js	https://34.64.188.58/cgi-bin/luci/;stok=/login
Possible status code response	status code: 200	**Not Found** Status code: 404	**Internal Server Error** Status code: 500
Model output	status code: 200	**Unauthorized** Status code: 401	**Internal Server Error** Status code: 500

Figure 13 – *GAN & BERT model response on unseen request data.*

III.4.2. Result of the Effectiveness of GAN-AIIPot

Evaluating a honeypot requires a combination of quantitative and qualitative metrics to provide a comprehensive assessment of its effectiveness. There are several metrics that can be used to evaluate the effectiveness of a honeypot, including:

The number of requests and diversity of attackers: the number of requests is a metric that measures the total number of requests captured by the honeypot. A higher number of requests can indicate a honeypot's attractiveness to attackers, but it doesn't necessarily mean that useful information was extracted from the attackers. The diversity of attackers is a metric that measures the number of unique

attackers that interact with the honeypot. A honeypot that attracts a diverse range of attackers can provide valuable intelligence about emerging threats and attack trends. The GAN & BERT model was deployed on the web server for 20 days. Table III summarizes the total number of requests the server received during this time and the number of unique IP addresses observed. Over the 20-day period, GAN-AIIPot captured 9,235 requests from 1,287 distinct IP addresses. This performance significantly surpasses AIIPot, which recorded 6,235 requests from 987 IP addresses, and Firmpot, which captured 2,896 requests from 687 IP addresses.

Table III – *Times of experimentation for each model.*

Method	Total number of requests received / total number of unique IP addresses observed
GAN-AIIPot	9,235/1,287
AIIPot	6,235/987
Firmpot	2,896/687

Session length: known as attack duration, this metric measures the length of time that an attacker spends interacting with the honeypot during a single session. Throughout the session, when an attacker sends a request, the honeypot generates a corresponding response, and the interaction concludes, resulting in a session length of 1. The duration of the session directly influences the perception of the honeypot as a genuine IoT device, with longer sessions leading to a higher likelihood of the attacker perceiving it as such. A longer duration of attacks can also indicate that the attacker was attempting to bypass the honeypot's defences, which can provide valuable information about the attacker's tactics and techniques. Therefore, session length is regarded as a key metric for evaluating the performance of honeypot deception, as well as the effectiveness of the learning process. Figure 14 shows the evaluation of the duration of the session of the proposed honeypot compared to the honeypot proposed by AIIPot and Firmpot.

We observe that the proposed honeypot captures the attention of the attacker in terms of session length. The attacker stays in session with the attacker for a long time compared to AIIpot and Firmpot. This confirms the robustness and effectiveness of the learning process. The proposed honeypot keeps 18% of the attacker in session length greater or equal to 7 but also reduces the percentage of session length 1 which goes from 54% and 62% that AIIPot and Firmpot obtain, respectively, to 48%.

Merely assessing the duration of interactions is insufficient to determine the comparative effectiveness of this approach in comprehending the behaviours of

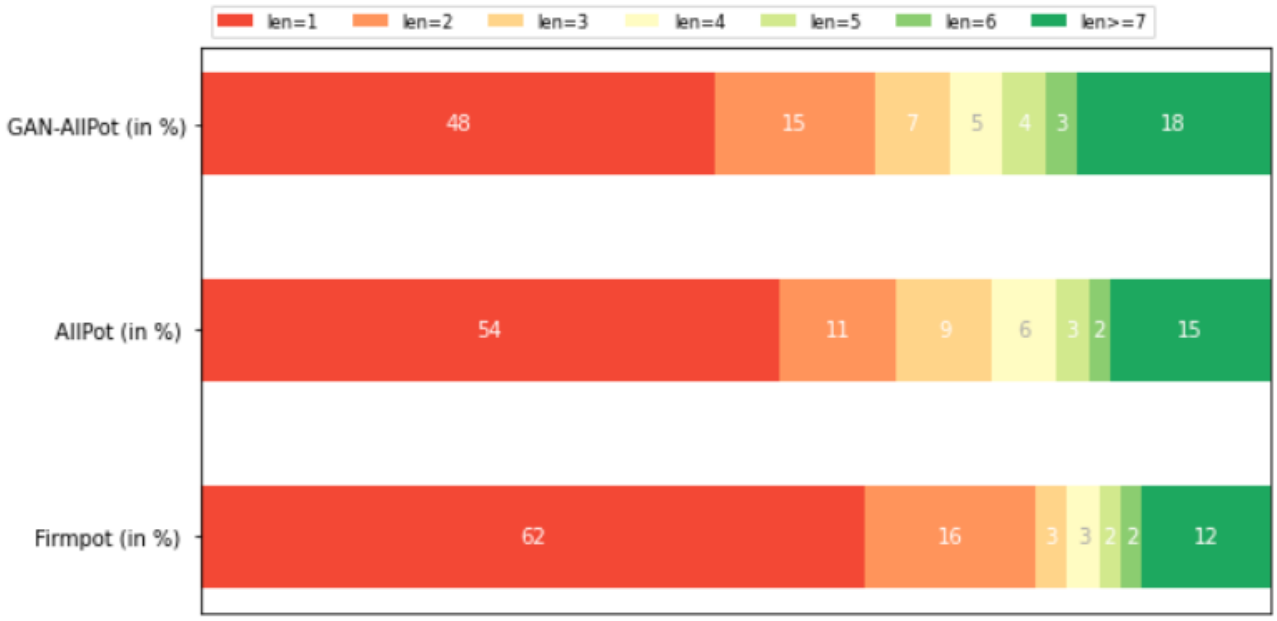


Figure 14 – *Session length comparison between attackers and honeypots (in percentage).*

attackers when compared to alternative methods. The goal of a honeypot is not necessarily to prolong the session length but to gather information about attacker techniques and behaviour to improve overall security. The significance of the information gathered from these interactions outweighs the importance of their duration. That's because attackers could only interact more because the honeypot is more verbose. Therefore, prioritizing the measurement of the extraction of valuable information from attackers becomes crucial in validating the duration of the attacker's session with the honeypot.

Data captured: A common metric for assessing the effectiveness of a honeypot based on session length is the amount of data (in bytes) transmitted by the attacker during each interaction within a session. This metric measures the amount and type of data captured by the honeypot. This data can be utilized to enhance incident response processes and create new security policies. Figure 15 shows that the higher the duration of the attacks, The larger the quantity of information transmitted. This makes sense because, during a session, more detailed requests typically result in a larger amount of information being transmitted. Sending more specific requests means that the attacker wants to know more about the target device. It also means that the attacker trusts the target device. Overall, from the boxplot in Figure 15 we can conclude that attackers who spend more time in GAN-AIIPot are those who trust the honeypot.

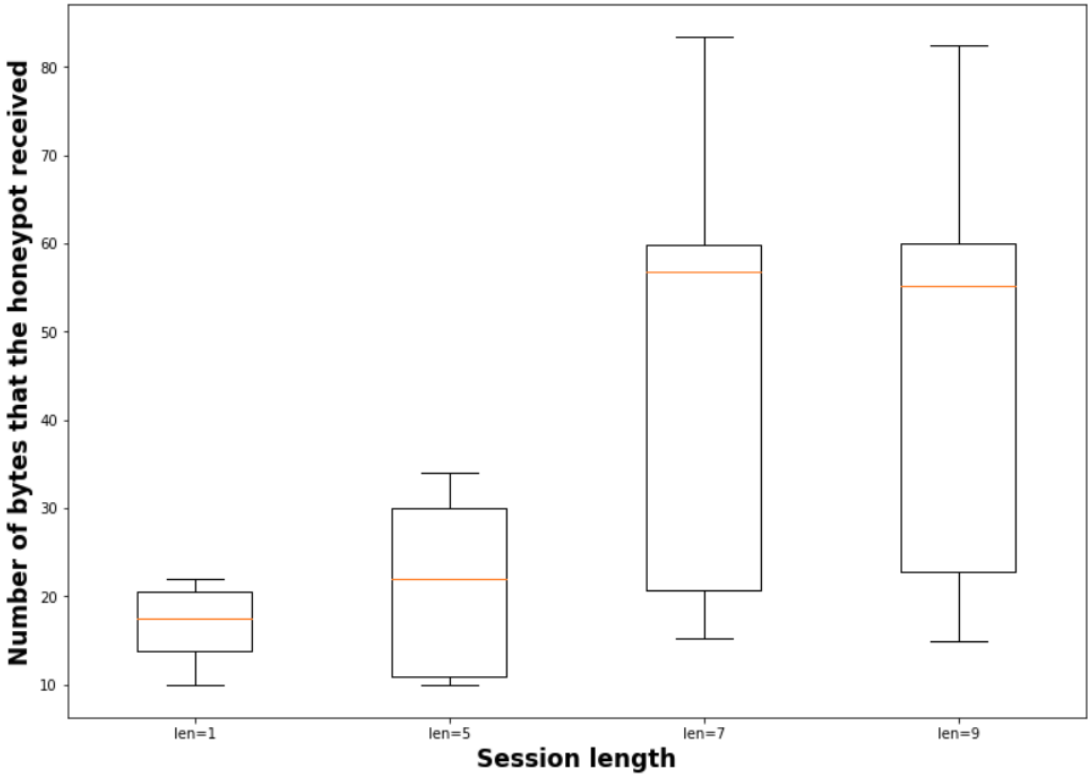


Figure 15 – *The amount of information received by the proposed honeypot per session length.*

Numbers and types of attacks: This metric assesses the effectiveness of a honeypot by tracking the different types of attacks it attracts. It helps in identifying common attack methods and developing new defenses against them. Using the SVM classifier model trained for the request evaluator module, we were able to measure the percentage of attacks captured by GAN-AIIPot. The results, shown in Figure 16, highlight the honeypot's effectiveness in detecting and capturing a substantial variety of attacks. Notably, we recorded a capture rate of about 35% for Denial of Service (DoS) attacks and around 70% for Remote to Local (R2L) attacks, which include activities like password guessing, database queries, and configuration changes. Furthermore, our honeypot successfully detected User to Root (U2R) attacks as well as probing attacks. These findings provide valuable insights into the behaviour of attackers targeting our system.

Cost-effectiveness: GAN-AIIPot is cost-effective because it is quicker and easier to deploy than deploying physical honeypots, as it is implemented entirely in software.

III.5. Analysis and Discussions

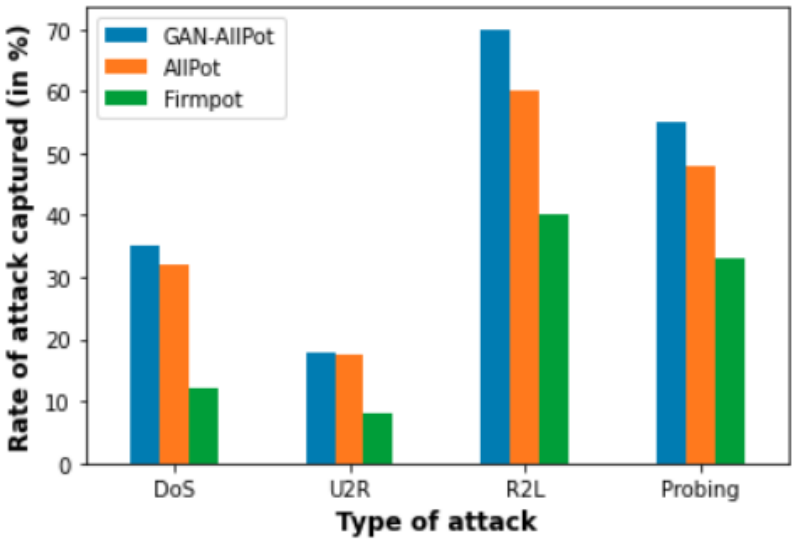


Figure 16 – *The count of attacks detected by the honeypot categorized according to the NSL-KDD dataset.*

In this section, we will analyze and discuss the findings from our contribution into the security challenges and countermeasures for IoT devices. We will go into the implications of probing attacks, evaluate the effectiveness of various machine learning techniques and cyber deception strategies, and explore the potential of innovative approaches like intelligent-interaction honeypots. Our goal is to provide a comprehensive understanding of the current state of IoT security and highlight the areas where further research and development are needed.

III.5.1. Analysis

The evaluation of our approach reveals its robust capabilities in addressing the core objectives of our thesis. Firstly, in the domain of identifying and analyzing common vulnerabilities in IoT devices, GAN-AIIPot has demonstrated significant effectiveness. The request evaluator module, powered by an SVM classifier, successfully identifies a wide range of attack types, including DoS, R2L, U2R, and probing attacks. The large number of requests and the diversity of attackers interacting with GAN-AIIPot, as observed from the collected data, indicate that the honeypot is highly attractive to various attack patterns. This comprehensive engagement provides a detailed understanding of IoT vulnerabilities, crucial for enhancing device security. Furthermore, the development of ML models for detecting cyber threats targeting IoT devices is showcased through the performance of our BERT and GAN models. The evaluation results indicate that the GAN & BERT model outperforms the standalone BERT model in terms of accuracy and conver-

gence speed. The learning curves for different fusion methods highlight that the combined model achieves higher accuracy and faster convergence, essential for real-time threat detection and response. This integration of ML models not only enhances detection capabilities but also supports adaptive responses to evolving cyber threats.

Additionally, the implementation of cyber deception techniques for misleading and capturing attackers has been validated through the metrics of session length and engagement. GAN-AIIPot effectively keeps attackers engaged for extended periods, with longer session lengths indicating successful deception. The adaptive response mechanism, guided by MDP, ensures realistic interactions that further deceive attackers, leading to the capture of detailed attack data. These results confirm the efficacy of combining ML with cyber deception to enhance IoT security. Overall, GAN-AIIPot surpasses initial expectations in multiple areas, affirming the potential of this approach in maintaining the security of IoT devices through innovative deception and adaptive mechanisms.

The originality of GAN-AIIPot lies in its novel integration of GAN and BERT models to create a dynamic and adaptive honeypot that effectively engages and deceives attackers. This approach significantly enhances the deception capabilities compared to traditional methods, offering a more resilient and responsive defense mechanism for IoT devices. The re-utilizability of GAN-AIIPot is a key strength, as its modular and scalable design allows for easy deployment across various IoT environments. This flexibility ensures that the framework can be adapted to different network configurations and security needs, making it a versatile tool for ongoing IoT security. By continuously learning from interactions and updating its strategies, GAN-AIIPot maintains its effectiveness in the face of evolving threats, ensuring long-term protection and valuable intelligence gathering.

III.5.2. Discussions

The findings from our evaluation of GAN-AIIPot highlight several critical aspects and implications for IoT security, setting it apart from existing work in significant ways.

Firstly, GAN-AIIPot's adaptive deception capabilities mark a substantial advancement over traditional honeypots. The integration of GAN with BERT allows for the generation of realistic and dynamic responses, effectively engaging attackers and keeping them deceived for longer durations. Unlike static honeypots, GAN-AIIPot continuously learns and adapts from interactions, making it more resilient against sophisticated attackers who might otherwise identify and evade less dy-

namic systems. In terms of scalability and cost-effectiveness, GAN-AIIPot offers a clear advantage. The virtual implementation ensures that it can be easily deployed and scaled across various IoT environments without the extensive resource demands associated with physical honeypots. This flexibility makes GAN-AIIPot a practical solution for large-scale IoT deployments, reducing both setup and maintenance costs while maintaining high performance.

The integration of ML models, particularly the combination of BERT and GAN, provides a significant boost in honeypot effectiveness. This dual-model approach not only enhances response accuracy but also allows the system to learn and adapt in real-time. This continuous learning capability is crucial for keeping up with rapidly evolving cyber threats, offering a more proactive defense compared to the reactive nature of traditional security measures. The data captured by GAN-AIIPot offers comprehensive insights into attacker behavior and tactics. This information is invaluable for security professionals aiming to understand and counteract emerging threats. By analyzing interaction patterns, GAN-AIIPot contributes to a more informed and proactive security posture, moving beyond the limitations of traditional reactive approaches. However, there are challenges and limitations to consider. One notable challenge is maintaining low latency in response times, which is essential for ensuring realistic interactions. Additionally, while GAN-AIIPot excels in deception, there is a constant need for updates and improvements to the ML models to stay ahead of increasingly sophisticated attackers. Ensuring the honeypot remains undetected by attackers who might employ advanced detection techniques is an ongoing challenge.

Comparatively, existing honeypots like Firmpot [75], IoTCandyJar [73] and Chameleon [74], while effective in their right, lack the dynamic and adaptive capabilities of GAN-AIIPot. IoTCandyJar relies on a Markov Decision Process model that takes time to learn and adapt, making it less responsive in real-time scenarios. Chameleon, although self-adaptive, depends heavily on initial configuration and manual updates to maintain effectiveness. Firmpot as a framework used firmware images and ML to learn the behaviors of IoT devices for intelligent interaction faced challenge on capturing most of well-know attacks. GAN-AIIPot, with its continuous learning and adaptive response mechanism, addresses these limitations, offering a more robust and scalable solution. This contribution represents a significant advancement in IoT security through the innovative use of ML and cyber deception. Its adaptive, scalable, and data-driven approach provides a valuable tool in the fight against IoT cyber threats.

III.6. Conclusion

In this chapter, we focused on the vulnerabilities of IoT devices, which are increasing due to the growth and variety of these devices. Traditional security measures often fail to protect IoT devices effectively, allowing attackers to bypass defenses and steal information.

To address this issue, we introduced AIIPot, an advanced honeypot designed to counter probing attacks on IoT devices. AIIPot uses intelligent adaptive interactions, combining machine learning techniques like BERT and GAN models with RL. This approach allows AIIPot to interact with attackers in real-time and capture valuable attack data and discover new vulnerabilities on IoT. We began by discussing the challenges in IoT security and explaining the phases and techniques of probing attacks. We emphasized the need for effective honeypot strategies and provided background on BERT, GAN, and RL to understand the techniques used in AIIPot. Next, we detailed AIIPot's architecture, including its components: the honey-chatbot, req/res database, and request evaluator. These components work together to create a sophisticated deception mechanism that responds accurately to attacker requests. The BERT and GAN models help generate responses that mimic real IoT device behavior, while RL ensures continuous improvement of the honeypot's effectiveness.

Our experiments showed that AIIPot outperforms traditional honeypots. The combination of GAN and BERT improved response accuracy and engagement duration, leading to the capture of more detailed attack data and discovered common vulnerabilities used by attacker on IoT device. AIIPot's real-time adaptation makes it a strong defense against sophisticated attacks and expansion of IoT devices thus vulnerability type, providing better protection for IoT devices.

IV

CHAPTER

Contribution to Adaptive Learning for IoT Security: A Hybrid Recommender System Approach

CONTENTS

IV.1 - Introduction 74
IV.2 - Preliminaries 75
IV.3 - Adaptive Learning-Based Hybrid Recommender Deception System . 77
IV.4 - Evaluation and Results 89
IV.5 - Analysis and Discussions 93
IV.6 - Conclusion 96

IV.1. Introduction

The rapid growth of IoT has connected everything from household appliances to industrial machinery, offering many benefits to our daily lives. However, this increased connectivity also brings notable security risks, making IoT devices prime targets for cyberattacks. These vulnerabilities can lead to unauthorized access, data breaches, or disruptions in operations, particularly in critical industries like healthcare, manufacturing, and transportation. While traditional security approaches such as software updates and patching are crucial, they are often reactive and may not provide immediate protection due to delays in patch release or deployment, leaving devices exposed. Additionally, proactive security measures are not always 100% effective in stopping attackers before they exploit system vulnerabilities.

IoT devices often share vulnerabilities due to similarities in vendors, firmware, or software, which skilled attackers can exploit. Considering that honeypots for

reconnaissance, like the one described in Chapter III, cannot capture all intrusions, and that some attacks come from internal sources, this presents a significant challenge: effectively concealing real vulnerabilities from attackers. Drawing inspiration from recommender systems, our approach implements a proactive deception strategy that offers attackers alternative, believable vulnerabilities. This tactic diverts their attention, protecting real devices by engaging them with these decoy vulnerabilities.

We propose an innovative hybrid recommender system that uses adaptive learning to combine the strengths of collaborative filtering (CF) and content-based filtering (CBF). This system goes beyond just detecting vulnerabilities; it also predicts potential vulnerabilities by analyzing both previous exploit behaviors and the inherent characteristics of vulnerabilities. The hybrid model continuously updates its recommendations by learning from real-time interactions, greatly improving its predictive accuracy and adaptability in an evolving threat environment.

This chapter presents a deception framework that utilizes an advanced hybrid recommender system with adaptive learning to strengthen IoT security. By integrating CF's ability to recognize patterns with CBF's focus on detailed attribute analysis, the system effectively detects and recommends deceptive vulnerabilities. These decoys are crafted to resemble real vulnerabilities in terms of credibility, complexity, and attractiveness, thereby safeguarding network integrity while gathering valuable insights into attacker tactics and tools.

The chapter will be structured as: We present the background of our contribution and necessary preliminaries to build up our contribution in section IV.2. Then in section IV.3 details in deep our proposed approach. Evaluation, result and comparison with the existing approach are presented in section IV.4 and section IV.6 summarise the key finding and the future work.

IV.2. Preliminaries

IV.2.1. Vulnerability on IoT Devices

The landscape of IoT security has evolved significantly due to the rapid growth and widespread use of IoT devices across various industries. Initially, IoT security focused on basic measures such as network protection. However, as IoT technology expanded into critical areas, the need for more sophisticated security strategies became unavoidable. These advanced methods aim to address the unique vulnerabilities introduced by the vast interconnectivity and diversity of IoT devices, high-

lighting the complexity of securing large networks against an ever-changing array of cyber threats.

The Common Vulnerability Scoring System (CVSS) plays a crucial role in this area by offering a standardized framework for assessing and prioritizing IoT vulnerabilities based on their severity and potential impact. This scoring system is key to creating targeted security responses, ensuring that the most critical vulnerabilities are addressed first. Recent research has explored the strategic use of fabricated vulnerabilities to mislead and redirect attackers. These studies emphasize the challenges of effectively using fake vulnerabilities—while they can be powerful tools for diversion, they also present difficulties, such as maintaining their credibility and managing the resources needed to maintain them.

Modern cybersecurity strategies increasingly incorporate advanced deception techniques. What began with basic honeypots has developed into complex deception architectures that employ a range of decoys and artificially generated data to actively engage and mislead attackers. This proactive approach goes beyond simply detecting intrusions; it interacts with attackers, steering them away from real targets, while gathering valuable intelligence about their tactics and tools. This shift towards a more active defense significantly improves the ability to create stronger, more effective security measures.

At the same time, recommender systems have become increasingly prominent in the field of cybersecurity.

One of the most innovative advancements in recommender systems is the development of hybrid systems, which combine the data-driven insights of Collaborative Filtering (CF) with the attribute-specific analysis of Content-Based Filtering (CBF). These hybrid systems leverage both historical interaction data and specific item attributes to offer precise and contextually relevant recommendations.

IV.2.2. Recommender Systems

Traditionally used in commercial settings to enhance user experience, recommender systems are now being adapted for cybersecurity applications. These systems play a key role in filtering and prioritizing security-related information, improving the detection and prevention of cyberattacks. Pawlicka et al. [119] examined how recommender systems can help manage information overload and improve decision-making in security operations centers. Meanwhile, Huff et al. [120] introduced a recommender system designed to automatically identify a minimal candidate set of common software product enumerators for software names, which helps enhance vulnerability identification and alert accuracy. However, ex-

isting research has not fully explored the use of recommender systems to propose a deceptive system.

Hybrid recommender systems, which merge CF and CBF, address the limitations of each method when used individually. CF is useful for identifying patterns based on user interaction data but struggles with the cold start problem, while CBF offers precise recommendations based on item attributes but may miss complex user preferences [121, 122]. Widayanti et al. [123] demonstrated how hybrid systems combining these methods can improve the accuracy and relevance of security recommendations in IoT contexts. Our research extends this approach by incorporating adaptive learning mechanisms, enhancing the system's ability to respond to evolving threats in real time.

IV.3. Adaptive Learning-Based Hybrid Recommender Deception System

This section presents an Adaptive Learning-Based Hybrid Recommender System (AL-HRS) designed to improve IoT security by implementing proactive deception. The system identifies existing vulnerabilities and dynamically recommends additional deceptive vulnerabilities by analyzing real-time attacker behavior and historical exploit data.

IV.3.1. System Design and Architecture

The architecture of AL-HRS includes three main components: the VARIoT database, the hybrid recommender system, and the deception framework. The VARIoT database holds extensive, up-to-date information on IoT vulnerabilities and exploits, reflecting the latest security developments. The hybrid recommender system merges CBF and CF, enhanced by adaptive learning algorithms, to suggest deceptive vulnerabilities. These suggestions are executed through the deception framework, which deploys controlled environments or honeypots to simulate the suggested vulnerabilities.

IV.3.2. Algorithms Used

AL-HRS utilizes a combination of CF and CBF, enhanced by adaptive learning algorithms. The system incorporates static features like vulnerability and device attributes (e.g., type, severity, manufacturer, and firmware version), along with dynamic features like exploit patterns (e.g., historical exploit attempts, frequency,

and recency). The integration of CF and CBF is achieved through a weighted sum approach, dynamically adjusted by an adaptive learning mechanism, specifically a Markov Decision Process (MDP).

CF analyzes interaction data from different attackers to predict vulnerabilities that similar attackers may exploit. In IoT security, this data includes exploit attempts on devices, offering insights into vulnerabilities commonly targeted by specific attackers. This helps forecast potential future targets.

CBF, on the other hand, focuses on item attributes (vulnerabilities and devices) to make recommendations, based on properties that have attracted attacks in the past. For IoT security, this includes the severity, exploitability, and impact scores (from CVSS), along with device-specific details like firmware version, type, and manufacturer. This allows for recommendations based on vulnerability similarities that might draw attacker interest.

The integration of CF and CBF is handled through a weighted sum approach, dynamically adjusted by an adaptive learning mechanism. The recommendation score for a vulnerability on a device is calculated as follows:

$$\text{Score}(v_j, d_i) = \theta_{CF} \cdot \text{CF_Score}(v_j, d_i) + \theta_{CBF} \cdot \text{CBF_Score}(v_j, d_i), \tag{IV.1}$$

where:

- θ_{CF} and θ_{CBF} are the adaptive weights assigned to the outputs from CF and CBF, respectively;
- $\text{CF_Score}(v_j, d_i)$ measures the relevance of a vulnerability based on interaction patterns of attackers with similar profiles;
- $\text{CBF_Score}(v_j, d_i)$ reflects relevance based on the intrinsic attributes of vulnerabilities and device profiles.

The closed-form definitions of CF_Score and CBF_Score are given as follows:

$$\text{CF_Score}(v_j, d_i) = \frac{\sum\limits_{k \in N(d_i)} \frac{f_d(d_i) \cdot f_d(d_k)}{\|f_d(d_i)\| \cdot \|f_d(d_k)\|} \cdot r_{k,j}}{\sum\limits_{k \in N(d_i)} \frac{f_d(d_i) \cdot f_d(d_k)}{\|f_d(d_i)\| \cdot \|f_d(d_k)\|}} \tag{IV.2}$$

and

$$\text{CBF_Score}(v_j, d_i) = \mathbf{w_v} \cdot f_d. \tag{IV.3}$$

where:

- v_j represents a vulnerability.

- d_i denotes a device in the IoT ecosystem.
- $N(d_i)$ is the set of neighboring devices similar to d_i, based on interaction or feature similarity.
- $f_d(d_i)$ is the feature representation of device d_i, capturing its characteristics.
- $f_d(d_k)$ represents the feature vector of a neighboring device d_k.
- $r_{k,j}$ denotes the interaction score between the attacker and vulnerability v_j on device d_k.
- $\mathbf{w_v}$ is a weight vector that represents the importance of different features in determining the relevance of a vulnerability.
- f_d is the feature representation of the device and vulnerability, incorporating intrinsic properties that influence the recommendation.

This hybrid approach allows AL-HRS to leverage both CF and CBF, enhancing its ability to create effective deception strategies for IoT security. The adaptive learning mechanism, specifically an MDP, continually fine-tunes θ_{CF} and θ_{CBF} based on real-time feedback to maximize deception effectiveness.

An MDP is a mathematical framework for decision-making in environments where outcomes are partly random and partly controlled by a decision-maker. In reinforcement learning (RL), MDPs are widely used to determine optimal policies—sequences of decisions that achieve the best possible outcomes. At each step in an MDP, the system's state changes in response to the actions taken, with transition probabilities determining the likelihood of moving from one state to another. The objective of an MDP is to identify a policy that maximizes expected rewards over time, taking into account uncertainties in state transitions and rewards. In this context, RL is used to iteratively improve decision-making by learning from interactions with the environment.

This section explores the creation of a deception defense strategy using the capabilities of an adaptive learning-based hybrid recommender system (AL-HRS) for IoT security. The strategy misleads attackers by dynamically presenting them with deceptive vulnerabilities, thereby protecting real devices.

IV.3.3. Overview

Figure 17 illustrates the system's architecture, designed to seamlessly integrate with existing IoT networks. It comprises three core components: the VARIoT database, AL-HRS, and the deception framework.

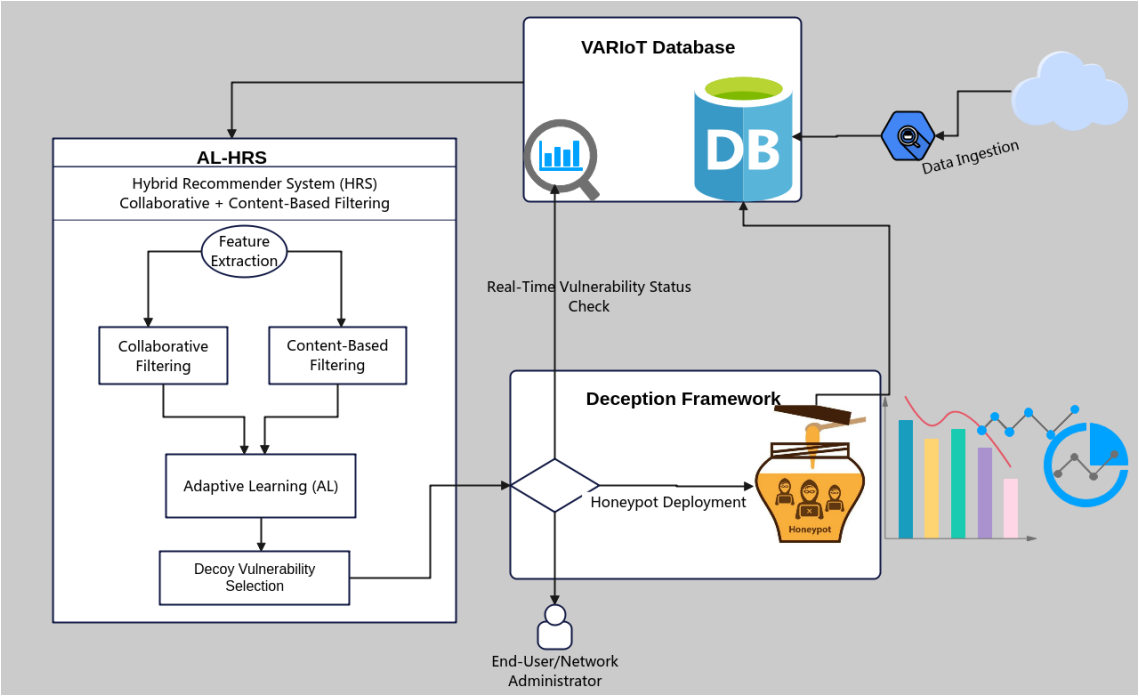


Figure 17 – *System architecture overview.*

- **VARIoT database:** This database serves as the backbone of the system, containing extensive data on IoT vulnerabilities and exploits [124]. Continuously updated, it reflects the latest security findings and trends in IoT vulnerabilities, sourced from databases like the Common Vulnerabilities and Exposures (CVE) and the National Vulnerability Database (NVD) [125]. The real-time updates ensure that the system's recommendations are based on the most current data.

- **AL-HRS:** At the heart of the system is AL-HRS, which analyzes vulnerability data from the VARIoT database along with real-time feedback on attacker behavior and interactions with decoy systems. The system employs a combination of CF and CBF, enhanced by adaptive learning algorithms. These algorithms adjust the recommendation logic based on the effectiveness of past decoys, optimizing future recommendations to keep them compelling to attackers while avoiding predictability.

- **Deception framework:** The AL-HRS-generated recommendations are executed through the deception framework, which deploys controlled environments or honeypots that simulate the vulnerabilities suggested by AL-HRS. These environments closely mimic real IoT device vulnerabilities, including simulated responses to attack attempts. This diverts attackers from real tar-

gets while collecting intelligence on attack methods, which is then fed back into AL-HRS to improve future recommendations.

IV.3.4. Detailed Structure of VARIoT Database

The VARIoT database is central to our approach. It is an online repository offering a comprehensive collection of IoT vulnerabilities and exploits [124]. Its role is to accurately and consistently profile IoT device vulnerabilities, which is essential for the system to recommend plausible deceptive vulnerabilities alongside real ones. The database's depth provides a solid foundation for simulating real-world IoT security scenarios. As shown in Figure 18, each entry includes IoT device characteristics (e.g., manufacturer/vendor, device type, operating system, firmware version, and network settings), vulnerability characteristics (e.g., severity, description, and impact), and exploit data (e.g., the exploit request/response).

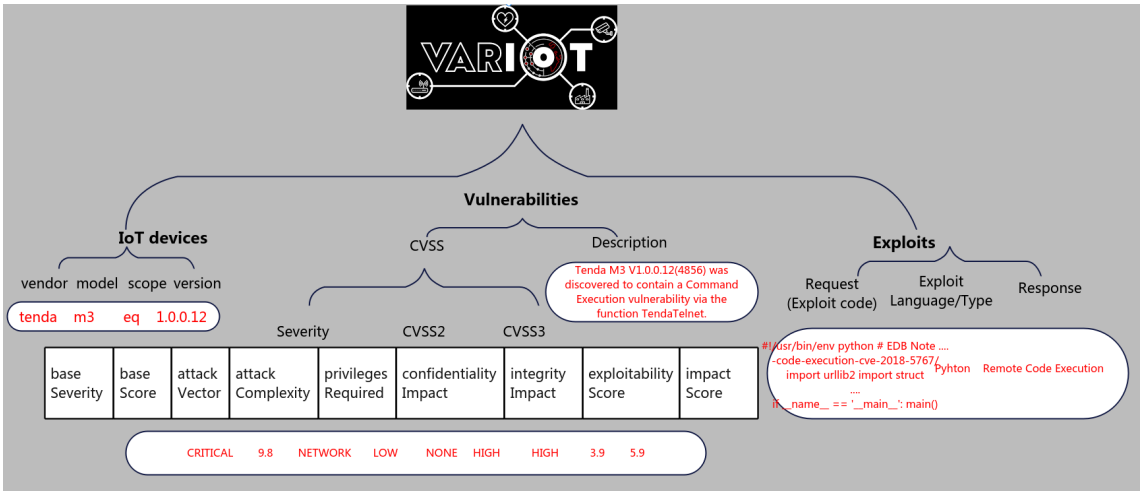


Figure 18 – *VARIoT database entries.*

IV.3.4.1. IoT Vulnerabilities

VARIoT is a dynamic repository that provides timely updates on IoT device vulnerabilities. It uses machine learning and natural language processing to merge information from diverse sources, creating detailed vulnerability descriptions and classifications [124]. As shown in Figure 18, the database records essential details for each affected device, including vendor, model, scope, and version. It also evaluates vulnerabilities based on the severity and type defined by the CVSS, as well as the current status of each vulnerability, allowing the recommender system to make precise and contextually relevant suggestions.

IV.3.4.2. IoT Exploits

This section of the VARIoT database catalogs publicly available exploits targeting IoT devices, enriched with details on their attributes and potential impacts. For repository development, this section mirrors the structuring found in Section IV.3.4.1. It plays a crucial role in our deception strategy by providing real-world data on exploit techniques, which, when analyzed, act as feedback from attackers. This feedback helps the system understand which vulnerabilities are most frequently exploited, enabling it to recommend the most effective deceptive vulnerabilities for specific IoT devices.

IV.3.5. Adaptive Learning-Based Hybrid Recommender System (AL-HRS)

AL-HRS is designed to dynamically recommend deceptive vulnerabilities to IoT devices, engaging potential cyber attackers. The system integrates CF and CBF, enhanced by an adaptive learning mechanism, evolving through real-time feedback on attacker behavior and deception effectiveness. AL-HRS uses a combination of static and dynamic features to form comprehensive vulnerability and device profiles.

- **Static Features** (CBF):
  - *Vulnerability Attributes*: Type, severity, related software/hardware;
  - *Device Attributes*: Manufacturer, device type, firmware version.
- **Dynamic Features** (CF):
  - *Exploit Patterns*: Historical exploit attempts, frequency, and recency.

IV.3.5.1. Collaborative and Content-Based Filtering Mechanisms

CF and CBF are two key methodologies used in AL-HRS to improve IoT device security by recommending deceptive vulnerabilities.

CF works by analyzing interaction data from various attackers to predict vulnerabilities that similar attackers might exploit. In the IoT security context:

- The interaction data includes exploit attempts on IoT devices, providing insights into vulnerabilities favored by different attackers;
- We identify patterns that indicate how attackers exploit similar devices or vulnerabilities, predicting which vulnerabilities are likely to be targeted next.

CBF focuses on item attributes (vulnerabilities and devices) to make recommendations, based on properties that have attracted attacks before. For IoT security, this includes:

- Attributes of vulnerabilities, such as their severity, exploitability, and impact scores (from CVSS), along with device-specific details like firmware version, type, and manufacturer;
- Recommending vulnerabilities based on their similarity to those previously exploited, considering specific attributes that might appeal to attackers.

Hybrid Recommendation Score: The integration of CF and CBF uses a weighted sum approach, dynamically adjusted by an adaptive learning mechanism. The recommendation score for a vulnerability v_j on a device d_i is calculated as follows:

$$\text{Score}(v_j, d_i) = \theta_{CF} \cdot \text{CF_Score}(v_j, d_i) + \theta_{CBF} \cdot \text{CBF_Score}(v_j, d_i),$$

where:

- θ_{CF} and θ_{CBF} are adaptive weights assigned to CF and CBF outputs, respectively;
- $\text{CF_Score}(v_j, d_i)$ quantifies the relevance of a vulnerability based on interaction patterns of attackers with similar profiles;
- $\text{CBF_Score}(v_j, d_i)$ measures relevance based on the vulnerability's and device's intrinsic attributes.

This hybrid approach allows AL-HRS to combine the strengths of both CF and CBF, improving its ability to formulate effective deception strategies for IoT security. The adaptive learning mechanism, particularly an MDP, continuously fine-tunes θ_{CF} and θ_{CBF} based on real-time feedback to optimize the deception's effectiveness.

Markov Decision Process: An MDP is a mathematical model used for decision-making in environments where outcomes are partly random and partly determined by a decision-maker [126]. MDPs are commonly used in RL to identify optimal policies—sequences of decisions that yield the best possible outcome [127]. In an MDP, the system's state can change with each action, with transition probabilities determining how likely it is to move from one state to another. The primary goal of an MDP is to identify a policy that maximizes the expected rewards over

time, accounting for uncertainties in state transitions and rewards. MDPs provide a structured framework for making a series of decisions, considering both immediate and future outcomes. This is crucial in environments where each decision affects future opportunities and risks. MDPs model the likelihood of transitioning from one state to another based on the selected action. The transition model defines the dynamics of the environment. In this context, RL is used to improve the decision-making policy through iterative adjustments. An RL agent learns from interactions with the environment, receiving rewards or penalties and adjusting actions based on past experiences to maximize cumulative rewards.

The MDP framework is vital for adaptive systems like AL-HRS, which adjusts its strategies dynamically to optimize deception effectiveness in IoT security by learning from past actions and their outcomes.

IV.3.5.2. Adaptive Learning for Weight Optimization

AL-HRS employs an RL framework to adjust the weights θ_{CF} and θ_{CBF} dynamically. These weights determine the balance between CF and CBF outputs in the hybrid recommendation model, optimizing the system's deception effectiveness against potential cyber attackers.

Problem Formulation AL-HRS interacts with an IoT ecosystem through an MDP, where agents (the recommendation system) engage with the environment (attackers). The system's objective is to learn a strategy that maximizes deception effectiveness through adaptive actions:

- **States** (s): Represent the current security status of the IoT environment, including detected attack patterns and known vulnerabilities;
- **Actions** (a): Involve selecting vulnerabilities to present as decoys, aiming to mislead attackers effectively;
- **Rewards** (r): Measure deception success, based on how well attackers are misled by the decoys;
- **Transition Probabilities** (T): Dictate the likelihood of moving from one state to another after an action, influenced by attackers' responses to deception strategies.

Reinforcement Learning Algorithm We apply Q-learning, an RL algorithm that optimizes the weights θ_{CF} and θ_{CBF}, refining the hybrid recommendation strategy:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a),$$

where γ is the discount factor, which prioritizes immediate rewards over future ones.

Reward Function The reward function is designed to reinforce actions that deceive attackers effectively:

- **Positive rewards** are assigned for actions that successfully engage attackers with decoys;
- **Negative rewards** are given for actions that fail to engage or result in the detection of decoys by attackers.

Update Rule The update rule modifies θ_{CF} and θ_{CBF} based on learned Q-values to continuously improve decision-making:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right], \tag{IV.4}$$

where α is the learning rate, which determines how quickly new information is integrated.

Exploration vs. Exploitation An ε-greedy strategy manages the trade-off between exploring new strategies and exploiting known effective strategies, optimizing learning while avoiding local minima.

Online Q-Learning Algorithm This system is designed to adapt in real-time to emerging threats and changing attacker behavior, continually refining and updating its strategies based on active interactions and feedback from the IoT environment. Algorithm 1 systematically outlines how AL-HRS operates. This focused reinforcement learning approach ensures that AL-HRS remains responsive and proactive, continually adjusting θ_{CF} and θ_{CBF} to improve deception capabilities against evolving cyber threats and maintain robust security within the IoT domain.

IV.3.6. Deception Framework

Our deception framework strategically integrates real and recommended vulnerabilities into the security profiles of IoT devices, as suggested by AL-HRS. This framework not only conceals real vulnerabilities but also engages potential attackers with carefully crafted decoys, directing them toward controlled environments that simulate actual device vulnerabilities. Before deploying deceptive measures, it is critical to ensure that the system can accurately distinguish between legitimate

Input: IoT device profiles $\mathbf{p}_i$, vulnerability data $\mathbf{v}_i$, historical exploit data $\mathbf{e}_i$
Output: Recommended vulnerabilities and their scores
begin
  Initialization:
  Initialize θ_{CF}, θ_{CBF} for CF and CBF;
  Load or initialize Q-table from previous sessions, if available;
  Configure parameters: learning rate α, discount factor γ, exploration rate ε;
  while *IoT system is active* **do**
    Collect current IoT device profiles, vulnerability and historical exploit data;
    Construct feature vector $\mathbf{x}_i$ for current device d_i from $\mathbf{p}_i$ (profiles), $\mathbf{v}_i$ (vulnerabilities), $\mathbf{e}_i$ (exploit history);

    Calculate CF and CBF scores for potential vulnerabilities;

    Obtain recommendation scores using current weights:

$$\text{Score}(v_j, d_i) = \theta_{CF} \cdot \text{CF_Score}(v_j, d_i) + \theta_{CBF} \cdot \text{CBF_Score}(v_j, d_i)$$

    Call Algorithm 2 to adjust θ_{CF}, θ_{CBF} based on reward signals;

    Deploy recommended vulnerabilities based on updated recommendation scores;

    Monitor attacker interactions and collect feedback;

    Validate and refine recommendations through simulations;
  end
end

Algorithm 1: AL-HRS: Adaptive Learning-Based Hybrid Recommender System

users and potential attackers. Drawing from established behavioral analytics and anomaly detection techniques [128, 129], our system implements advanced monitoring techniques to analyze access patterns and behavior. This ensures that normal user activities are not affected by deception tactics aimed at attackers, maintaining seamless IoT device operation while enhancing security.

Integration of Vulnerabilities When a real vulnerability is identified in a device, the system checks its current status against the VARIoT database. If the vulnerability has been resolved, the system either applies a security patch automatically or alerts network administrators for immediate action. This proactive approach en-

Input: Current state s_t, current Q-values $Q(s,a)$, learning rate α, discount factor γ, exploration rate ε
Output: Updated Q-values $Q(s,a)$, selected action a_t
while *receiving feedback from the IoT environment* **do**
  Observe current state s_t (current security context and attacker interactions);
  Select action a_t (choose weights to apply) using ε-greedy policy;
  if *random()* $< \varepsilon$ **then**
    Choose a random set of weights;
  end
  else
    Choose weights that maximize current Q-values:

$$a_t = \arg\max_a Q(s_t, a)$$

  end
  Apply weights, observe reward r_t and new state s_{t+1};
  Update Q-values:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t) \right]$$

  ;
  Optionally adjust ε to reduce exploration over time;
end

Algorithm 2: Online Qlearning for ALHRS

sures vulnerabilities are addressed quickly, protecting the network from potential exploits.

Deployment of Decoys Simultaneously, the system enhances the device's security profile by integrating recommended fake vulnerabilities. These decoys are presented alongside real vulnerabilities to create a more nuanced security layer that misleads attackers. Each deceptive vulnerability is paired with a corresponding honeypot, simulating the characteristics of the fake vulnerability. For example, if a vulnerability related to an open TCP port is suggested, a honeypot emulating an open port is deployed. This diverts attackers from real assets and allows monitoring and analysis of their attack methods in a risk-free environment.

Honeypot Operations When attackers target these fake vulnerabilities, they are redirected to honeypots. These honeypots [130] are sophisticated enough to convincingly engage attackers, thus protecting real systems and collecting valuable data on attack patterns and techniques. This intelligence is crucial for continuously refining deception strategies and updating AL-HRS recommendations.

Feedback Loop All honeypot interactions are logged and analyzed, with significant exploits added to the VARIoT database. This continuous feedback enriches the data used for future recommendations, ensuring that the deception framework remains dynamic and responsive to evolving cybersecurity threats. This cyclical process not only mitigates immediate threats but also strengthens IoT environments against future attacks.

IV.3.7. Use Case and Implementation

In this section, we provide a use case illustrating how the system enhances IoT security by combining accurate vulnerability assessment with strategic deception, turning potential weaknesses into opportunities for defense and intelligence gathering.

IV.3.7.1. System Integration and Workflow: Threat Model

The system is designed to integrate seamlessly with IoT networks while being robust against specific cybersecurity threats. Our threat model assumes a landscape where attackers continuously scan IoT devices to exploit known vulnerabilities. In response, the system leverages the VARIoT database, which is regularly updated with the latest vulnerability data. This data is essential for AL-HRS to function effectively. AL-HRS utilizes feedback from previous attacker engagements, including information on which vulnerabilities (decoys) were exploited and the methods used. This feedback refines the system's predictive models, using adaptive learning techniques to anticipate and simulate new, plausible deceptive vulnerabilities.

These recommended vulnerabilities are strategically incorporated into the deception framework, creating detailed, realistic profiles of device vulnerabilities. When attackers scan the network, they encounter these decoys, which lead them into honeypots simulating fake vulnerabilities. This approach neutralizes threats by diverting attackers from real targets and engaging them in a controlled environment. This not only reduces immediate risks but also collects valuable intelligence on attack methods, helping improve long-term defensive capabilities.

IV.3.7.2. Example Scenario: Securing a Surveillance Camera

Consider an IP camera connected to an IoT network, a common target for cyber attackers due to its access to sensitive visual data. Initially, the system integrates with the network and identifies any known vulnerabilities in the specific IP camera model. For example, it might detect a vulnerability related to *unsecured data transmissions*.

Vulnerability Analysis and Recommendation After identifying real vulnerabilities, the recommender system analyzes their characteristics—such as nature, severity, CVSS score, and device type. It then uses this analysis to select additional plausible vulnerabilities based on AL-HRS that could exist in similar devices but have not yet been exploited, such as *open TCP ports or firmware exploits*. This creates a comprehensive vulnerability profile for the IP camera, combining real and fake vulnerabilities.

Deception Strategy Implementation The vulnerability profile is embedded in a virtual representation of the IP camera within the deception framework. This virtual profile acts as a digital façade, displaying both real and fake vulnerabilities. Techniques like port mirroring or network traffic redirection ensure that attackers interact with the façade rather than the actual device. The system simulates network responses to reflect the fake vulnerabilities, such as responding to queries targeting these decoys.

Engagement with Attackers If a user identified as an attacker targets the recommended vulnerabilities, such as trying to exploit the *open TCP port*, they are redirected to a honeypot. The honeypot is carefully configured to emulate an IP camera with the specified vulnerabilities, creating the illusion of a successful breach. The system monitors this interaction, recording the attacker's tactics, tools, and method of exploitation. This provides valuable insight into current attack methodologies and helps identify specific threat actors.

At the same time, the real IP camera is secured by applying recommended patches or updates to fix its vulnerabilities. The honeypot not only diverts attackers but also acts as a learning tool, helping the system adapt to evolving attack patterns and ensuring continuous network security.

IV.4. Evaluation and Results

This section presents the results of a comprehensive evaluation of AL-HRS, designed to assess its effectiveness in various conditions and setups.

IV.4.1. Accuracy Metrics

As shown in Table 4, AL-HRS achieved an average precision of 92%, reflecting its ability to identify suitable deceptive vulnerabilities across a range of attack scenarios. The system's recall was 89%, showing its effectiveness in detecting ex-

Table 4 – *Comparison of accuracy metrics across different systems*

System	Precision	Recall	F1 Score
AL-HRS	92%	89%	90.5%
Hybrid CF+CBF	85%	82%	83.5%
CF Only	80%	78%	79.0%
CBF Only	78%	75%	76.5%

ploitable vulnerabilities in simulated IoT devices. The overall F1 score was 90.5%, indicating a well-balanced performance in deception-based recommendations. AL-HRS performed significantly better than other systems in this evaluation.

Table 5 – *Recommended vulnerabilities for smart home devices.*

Device Type	CVE ID	Recommended vulnerability	CVSS Exploitability
Smart Thermostat	CVE-2023-5002	Buffer Overflow in Communication Protocol	8.5
Smart Light	CVE-2023-5003	Firmware reversion attack	7.1
Smart Fridge	CVE-2023-5004	Cross-Site Scripting in Web Interface	6.9
Home Security Camera	CVE-2023-5005	Remote code execution via API	9.8
Smart TV	CVE-2023-5006	Denial of Service through Malformed Inputs	7.5
Home Assistant Hub	CVE-2023-5007	Eavesdropping vulnerability in voice commands	8.2
NAS Server	CVE-2023-5008	SQL injection in device management interface	9.0

Table 5 lists some vulnerabilities recommended by AL-HRS for each device type in a simulated smart home environment. The table specifies the type of vulnerability, the expected CVSS exploitability score, and the decoy's effectiveness in engaging potential attackers. These vulnerabilities act as decoys to mislead attackers and analyze their interaction patterns.

IV.4.2. Effectiveness of Deception

Figure 19 shows the evaluation of engagement times, which indicates how long attackers interact with decoys before halting their activities. Our results show that deceptive vulnerabilities recommended by AL-HRS successfully engaged 26% of

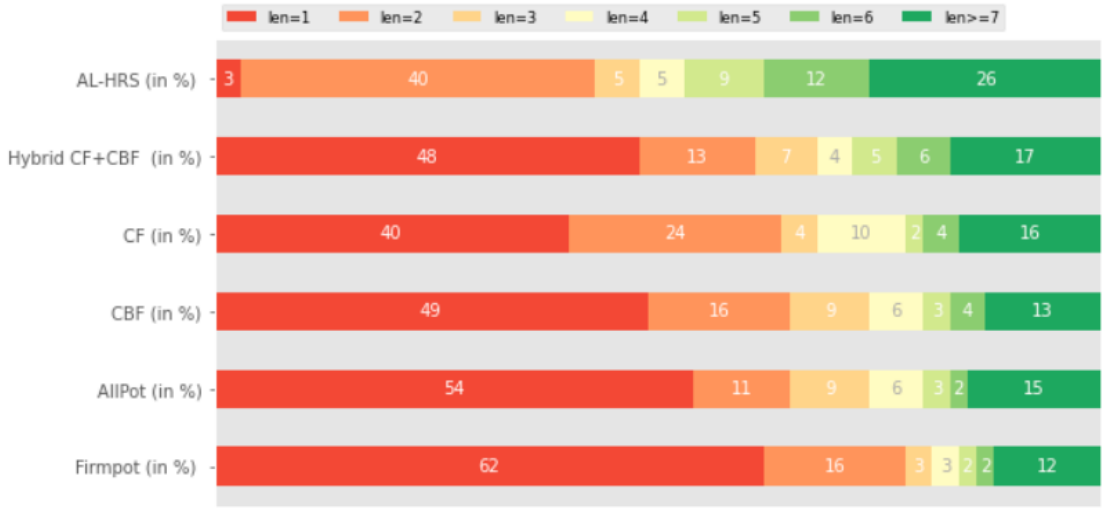


Figure 19 – *Comparison of attacker engagement times with recommended vulnerabilities by AL-HRS and existing method.*

attackers, with an average session length of 7 minutes. This was notably higher than traditional deception methods. Further analysis revealed variations in engagement times: shorter sessions lasted about 2 minutes, while longer sessions extended up to 12 minutes. This variability highlights the system's ability to adapt to different attacker strategies, maintaining engagement across a wide range of attack types.

Engagement time alone does not fully demonstrate whether our approach outperforms others in understanding attacker behavior. Therefore, we focused on the ability to extract actionable intelligence from attackers during their engagement. Our system demonstrated a high detection rate for well-known attacks, around 78%, as shown in Figure 21. Additionally, we analyzed the volume of information attackers transmitted during engagement. Figure 20 shows a clear correlation between engagement time and the volume of information sent by attackers. This trend is logical—longer sessions tend to involve more specific requests, increasing the amount of information exchanged. This indicates that attackers are more likely to trust and engage with the decoys, suggesting they believe in the authenticity of the recommended vulnerabilities. This correlation highlights the effectiveness of our system in engaging attackers and gathering valuable intelligence during these interactions.

Overall, AL-HRS outperformed other methods by at least 15% in key metrics such as engagement time and detection rates. This superior performance underscores the advantages of integrating adaptive learning and hybrid recommendation strategies, particularly in complex

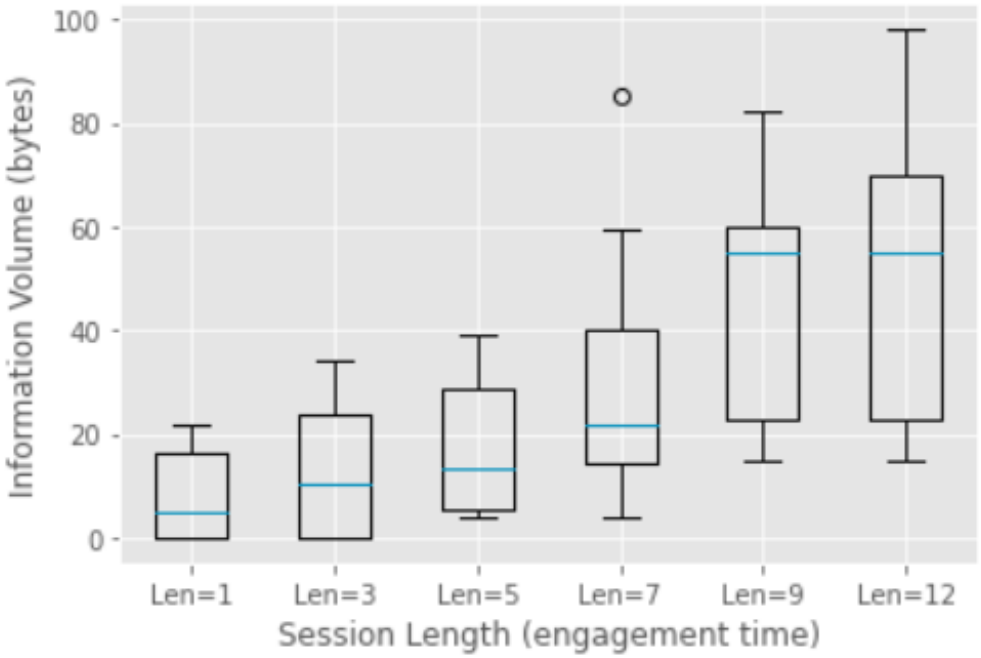


Figure 20 – *Correlation between engagement time and volume of information sent by attackers, indicating the effectiveness of the recommended vulnerabilities in extracting valuable intelligence.*

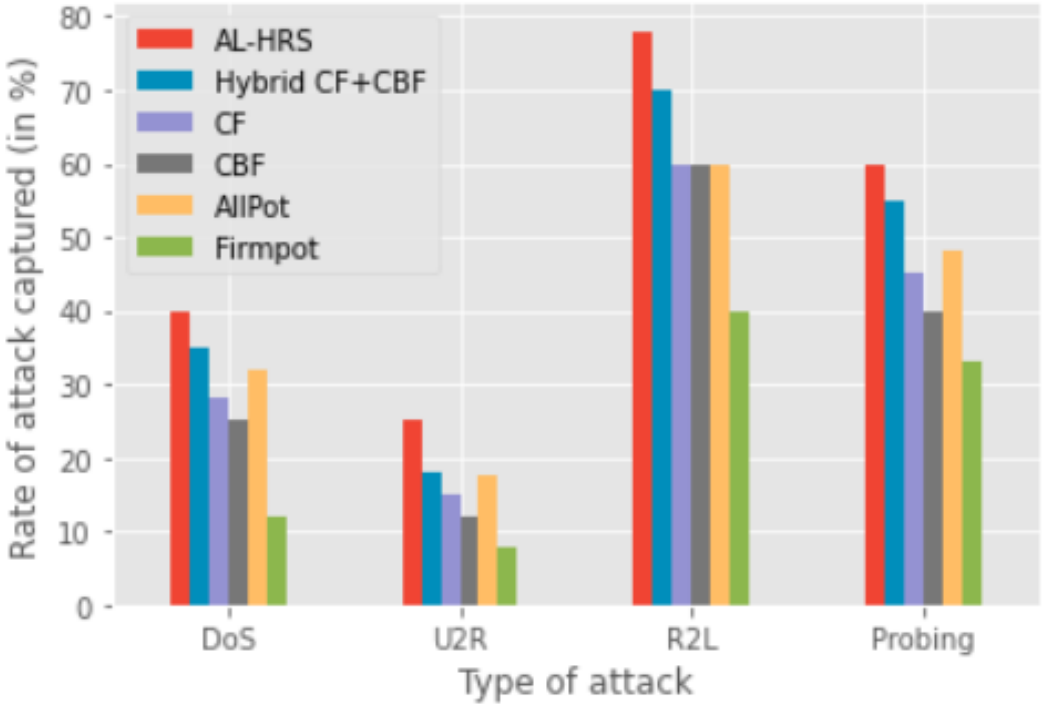


Figure 21 – *Detection rate of AL-HRS showing the system's effectiveness in identifying and responding to well-known attack types such as denial of service (DoS), probe, user to root (U2R), and remote to Local (R2L).*

, varied testing environments.

IV.4.3. Scalability and Response Efficiency

The system maintained a response time of under 2 seconds in 95% of cases, even as the network scaled to 100 devices. This quick response, combined with the system's ability to handle increased load without performance loss, demonstrates its scalability and efficiency. Moreover, AL-HRS successfully mitigated over 50% of advanced persistent threats and zero-day exploits, showing its adaptability to emerging threats.

Table 6 – *Evaluation of the attractiveness of the deception system.*

Method	**Total number of requests received / total number of requests targeting recommended vulnerabilities**
AL-HRS	9,235 / 5,287
Hybrid CF+CBF	7,235 / 1,917
CF only	6,235 / 1,087
CBF only	5,235 / 993
AIIPot [130]	6,235 / 987
Firmpot [75]	2,896 / 687

Over a one-month testing period, the system's effectiveness improved by 15%, reflecting its ability to learn and adapt based on attacker feedback and shifting attack patterns. During this period, the system captured 9,235 requests, with 5,287 targeting the recommended vulnerabilities. This performance far surpassed non-adaptive hybrid CF+CBF systems, CF-only, CBF-only, and AIIPot [130]. Table 6 shows that although the actual vulnerabilities attracted a large number of requests, the recommended vulnerabilities received an even higher number of requests. This demonstrates that the decoys effectively distracted attackers from real vulnerabilities. Interestingly, despite the higher number of requests for recommended vulnerabilities, fewer attackers targeted them than real vulnerabilities, indicating successful misdirection and confusion.

IV.5. Analysis and Discussions

IV.5.1. Analysis

The evaluation of AL-HRS shows its considerable potential in enhancing IoT security through proactive deception. The system was tested in various conditions, providing a comprehensive analysis of its effectiveness.

Firstly, AL-HRS achieved impressive accuracy metrics, with an average precision of 92%, a recall rate of 89%, and an overall F1 score of 90.5%. These metrics indicate that the system is highly proficient in identifying and recommending suitable deceptive vulnerabilities across different attack scenarios. Compared to traditional methods like CF-only, CBF-only, and non-adaptive hybrid approaches, AL-HRS demonstrated superior performance, reflecting its ability to effectively combine collaborative and content-based filtering with adaptive learning.

The effectiveness of AL-HRS's deception strategies was also evident. The system successfully engaged 26% of attackers, with an average session length of 7 minutes, significantly longer than traditional methods. This extended engagement time suggests that attackers were effectively misled by the recommended vulnerabilities, interacting with the decoys for longer periods. The analysis showed variability in engagement times, reflecting the system's ability to adapt to different attacker behaviors and sustain engagement across a broad spectrum of attack types. Moreover, the evaluation demonstrated the system's ability to gather valuable intelligence from attackers. There was a clear correlation between engagement time and the volume of information exchanged, indicating that longer sessions led to more detailed information being shared. This underscores the system's effectiveness in extracting actionable insights into attacker tactics and behavior, which is crucial for refining future deception strategies.

In terms of scalability and response efficiency, AL-HRS maintained a response time of under 2 seconds in 95% of cases, even as the network expanded to 100 devices. This quick response, combined with the system's ability to handle increased load without performance loss, demonstrates its scalability and efficiency. The system also mitigated over 50% of advanced persistent threats and zero-day exploits, showing its adaptability to new and emerging threats. Overall, the results suggest that AL-HRS outperforms existing methods in key metrics like engagement time, detection rates, and intelligence gathering. Its superior performance highlights the benefits of integrating adaptive learning and hybrid recommendation strategies in complex, varied environments.

The novelty of AL-HRS lies in its innovative combination of adaptive learning, CF, and CBF to create a robust and dynamic security solution for IoT networks. This approach improves the system's ability to detect and deceive attackers, while

also continuously evolving to address new threats. The reusability of this contribution is significant, as the framework can be adapted to various IoT environments and security contexts. The modular design of AL-HRS makes it easy to integrate into existing security infrastructures, making it a valuable tool for enhancing IoT security across different applications. By offering a scalable, adaptable solution, AL-HRS ensures long-term relevance and effectiveness in protecting IoT ecosystems from sophisticated cyber threats.

IV.5.2. Discussions

AL-HRS offers several important advantages and differences compared to existing IoT security methods. Traditional approaches, like CF-only or CBF-only methods, often struggle with issues such as the cold start problem and limited adaptability to new attack patterns. By combining CF and CBF with an adaptive learning mechanism, AL-HRS overcomes these challenges, providing a more robust and dynamic solution.

One of the main strengths of AL-HRS is its ability to continuously learn and adapt based on real-time feedback from attacker interactions. This adaptive learning capability enables the system to refine its recommendations and deception strategies, ensuring that it remains effective even as attack patterns evolve. In contrast, traditional methods typically rely on static models that may become outdated as new vulnerabilities and attack techniques emerge.

The hybrid recommender system also improves the accuracy and relevance of the recommended vulnerabilities. By combining CF's pattern recognition with CBF's detailed attribute analysis, AL-HRS can identify and suggest deceptive vulnerabilities that are both credible and appealing to attackers. This dual approach ensures that the decoys are not only believable but also effective in diverting attackers from real vulnerabilities.

Additionally, the MDP-based RL framework further distinguishes AL-HRS from existing methods. This framework allows the system to optimize its deception strategies by learning from the outcomes of past actions and associated rewards. The continuous adjustment of weights for CF and CBF outputs based on real-time feedback ensures that the recommendations remain accurate and effective over time.

However, there are some challenges and limitations to consider. One significant challenge is balancing engagement and detection. While the goal is to keep attackers engaged with decoys for as long as possible, it is also crucial to ensure that the decoys do not reveal too much information that could be used against the system

in future attacks. Achieving this balance requires careful design and continuous refinement of deception strategies.

Furthermore, implementing the system in a real-world environment may present challenges related to integration with existing security infrastructures and managing false positives. Ensuring that the system can seamlessly integrate with existing IoT networks and accurately differentiate between legitimate users and attackers is essential for its effectiveness.

Compared to existing methods, AL-HRS demonstrates significant improvements in key metrics like engagement time, detection rates, and the volume of information extracted from attackers. The system's ability to adapt to emerging threats, combined with its scalability and efficiency, makes it a valuable tool for enhancing IoT security through proactive deception.

In conclusion, AL-HRS represents a major advancement in IoT security by combining adaptive learning, hybrid recommendation strategies, and proactive deception. Its ability to learn from real-time interactions and continuously refine its strategies ensures that it remains effective against evolving cyber threats, providing robust protection for IoT networks.

IV.6. Conclusion

In this chapter, we explored the significant security challenges faced by IoT devices, emphasizing the importance of proactive and adaptive security measures. IoT devices, from smart home appliances to industrial sensors, are becoming increasingly prevalent, making them attractive targets for cyber attackers. The diversity and varying security levels of these devices create numerous vulnerabilities that attackers can exploit. Traditional security measures, such as software updates and patching, often fall short of providing real-time protection and struggle to keep pace with the rapidly evolving threat landscape.

To address these challenges, we introduced the Adaptive Learning-Based Hybrid Recommender System (AL-HRS), which enhances IoT security through a sophisticated deception framework. AL-HRS aims to protect IoT devices by misleading attackers with plausible yet fake vulnerabilities, thereby diverting their focus from real targets. The system has two key objectives: first, to identify and recommend deceptive vulnerabilities that closely mimic real ones, and second, to continually learn from interactions with attackers to refine its deception strategies. By integrating CF and CBF with adaptive learning, AL-HRS can identify and recommend deceptive vulnerabilities effectively. CF analyzes patterns in historical attack

data to predict which vulnerabilities attackers might target next, while CBF uses the intrinsic attributes of devices and vulnerabilities to generate relevant recommendations. The adaptive learning component, particularly the MDP, ensures that AL-HRS can dynamically adjust its recommendations based on real-time feedback, improving its ability to engage and mislead attackers.

Our evaluation demonstrated that AL-HRS significantly improves engagement times and the extraction of valuable intelligence compared to traditional methods. Attackers spent more time interacting with the decoys, providing deeper insights into their tactics and techniques. The system's ability to dynamically adjust its strategies based on real-time feedback ensures ongoing adaptability and resilience against evolving cyber threats. This continuous learning process not only enhances immediate security measures but also contributes to a more robust long-term defense strategy.

V

CHAPTER

Contribution to Advanced Security using Augmented Reality Honeypots: A Case Study for IoT Cameras

CONTENTS

V.1 - Introduction 98
V.2 - IoT Cameras and Augmented reality 99
V.3 - Adaptive Honeypot-based Augmented Reality for IoT Camera: AR-PotCam 101
V.4 - Evaluation and Results 114
V.5 - Analysis and Discussions 124
V.6 - Conclusion 127

V.1. Introduction

The rapid proliferation of IoT devices, particularly IoT cameras, has brought about transformative changes in various domains. These devices have revolutionized how we monitor and interact with the physical world, enabling seamless surveillance, remote monitoring, and enhanced user experiences. Concurrently, the advent of augmented reality (AR) has guided a new era of immersive and interactive visual experiences, blurring the boundaries between the physical and digital environments. In cybersecurity, honeypots have emerged as indispensable tools for detecting, analyzing, and mitigating potential threats. Honeypots are strategically deployed systems that mimic real targets to lure and observe malicious actors, offering invaluable insights into their tactics, techniques, and procedures.

The intersection of these three domains — IoT, cameras, AR, and honeypots — creates a unique avenue for exploration. While IoT cameras have become integral to modern surveillance and monitoring solutions, their ubiquitous presence and potential vulnerabilities, such as default, weak, and hard-coded passwords, make them susceptible to malicious exploitation. To protect these devices and networks against cyber threats, the need for innovative and effective security measures is essential. Traditional honeypots have long been instrumental in capturing network-based cyber threats, yet they struggle with inherent limitations. These limitations, ranging from attackers becoming increasingly adept at distinguishing between real and emulated environments to missing out on more sophisticated attacker behaviors, have prompted a paradigm shift in honeypot design. The motivation to incorporate AR and RL into our honeypot approach arises from recognizing that AR offers a unique solution to many of these challenges, while RL enhances intelligent decision-making capabilities making the system to adapt and keep the attacker into deception for long.

This chapter introduces an approach called **ARPotCam**: "Augmented Reality-based honeyPot for IoT Cameras." The key challenge addressed is designing a honeypot that convincingly emulates the behavior of an IoT camera while accurately reacting to camera control commands, closely mimicking a real device. Unlike traditional honeypots, which focus primarily on network traffic, this novel honeypot explores the unique combination of AR with IoT security. Additionally, the fusion of AR with IoT security is strengthened by the power of RL to adapt with the attacker interaction.

The chapter is organized as follows: Section V.2 offers foundational knowledge and examines the current literature on honeypots specifically designed for IoT cameras. Section V.3 details our methodology, including the technical implementation of the AR-based honeypot, our experimental setup, and evaluation methods. Section V.4 presents our experimental results, discusses their implications, and analyzes the strengths and limitations of our proposed approach. Finally, Section V.6 offers conclusions drawn from our findings and outlines directions for future research.

V.2. IoT Cameras and Augmented reality

V.2.1. IoT Cameras

IoT cameras are a critical component in the network of connected devices used for surveillance, monitoring, and security. These cameras provide real-time video feeds and remote access capabilities, enhancing situational awareness and security across various domains, including residential, commercial, and public spaces. Typically, IoT cameras consist of several key components: a camera sensor that captures video and images, a microcontroller or processor that processes the captured video data, a network interface that enables communication with other devices and servers over the internet, storage that can be local or cloud-based for recorded data, and firmware that controls the camera's functions and features.

Despite their benefits, IoT cameras are often vulnerable to cyber threats due to several factors. Weak passwords, which are often set to default values and easily guessable, make unauthorized access a common issue. Outdated firmware that lacks regular updates can leave cameras exposed to known vulnerabilities. Additionally, the lack of encryption in data transmitted by these cameras can make it susceptible to interception, leading to unauthorized access, data breaches, and privacy violations.

To analyze the security of IoT cameras, various mathematical models and frameworks can be used. One common approach is to model the network as a graph $G = (V, E)$, where V represents the set of devices (nodes) and E represents the connections (edges) between them. Security analysis involves evaluating the robustness of this graph against attacks, which can be quantified using metrics such as node degree, clustering coefficient, and network diameter.

V.2.2. Augmented Reality

AR technologies have rapidly evolved, transforming the way we perceive and interact with the digital world. Augmented Reality (AR) can be described as a real-time, direct or indirect view of the physical world, enhanced by the addition of virtual, computer-generated content. The objective of AR is to blend real and virtual elements, enriching the physical environment with digital information. These augmentations may vary from simple text overlays on real-world objects to virtual replicas of actual items integrated into a real setting. In cases where virtual objects are seamlessly inserted, the goal is to make the distinction between real and virtual elements indistinguishable. AR can also involve the removal of real objects by overlaying virtual content that blends with the surroundings, creating the illusion that the object is absent. AR is characterized by three key features:

(i) combining real and virtual elements, (ii) being interactive in real-time, and (iii) being registered in 3D, as outlined by Azuma [131].

An AR experience takes place in the minds of one or more participants, where AR technology creates artificial stimuli that lead individuals to perceive events that aren't actually happening. Elements that don't exist in the physical world are made to appear as though they do. Participants play an active role in shaping the AR experience, as their movements, actions, and behaviors directly influence how the system reacts. AR has a wide range of applications, such as in gaming, education, marketing, and healthcare. However, the intersection of AR and the security field, particularly in IoT security and honeypots, is still a largely untapped area of research [132].

In contrast, our research addresses these limitations by pioneering a deceptive IoT camera honeypot system based on AR techniques. AR-based solutions offer RL real-time interactivity, responsiveness, and the ability to seamlessly integrate virtual elements into the physical environment. This allows for a more immersive and convincing honeypot experience that can better engage attackers and provide more accurate insights into their behaviour.

V.3. Adaptive Honeypot-based Augmented Reality for IoT Camera: ARPotCam

This section presents our innovative approach, which aims to provide captivating and misleading video streams through the use of AR, originating from authentic IoT cameras. We operate under the assumption that potential attackers employ port scanning tools or specialized search engines like Shodan to identify vulnerable IoT cameras. The attackers' strategies encompass exploiting device vulnerabilities or employing password-cracking techniques, aiming to access sensitive private data, such as video streams, or establish malware for subsequent attacks. Furthermore, attackers may issue camera control commands to validate the authenticity of devices with high interactivity. Our primary objective revolves around the creation of a specialized honeypot designed exclusively for the context of IoT cameras. To accomplish this, we directly acquire the video stream from an authentic IoT camera. This camera's inherent ability to deliver a live video feed is directed towards an AR unit. Within this AR framework, we capitalize on the live video stream as the foundational element on which we seamlessly integrate or remove objects, crucial for achieving meticulous and precise deception. We specifically examine scenarios where the real IoT camera remains stationary. The underlying intention is to en-

hance the authentic video stream through augmentation, thus misleading potential attacks.

V.3.1. System Overview

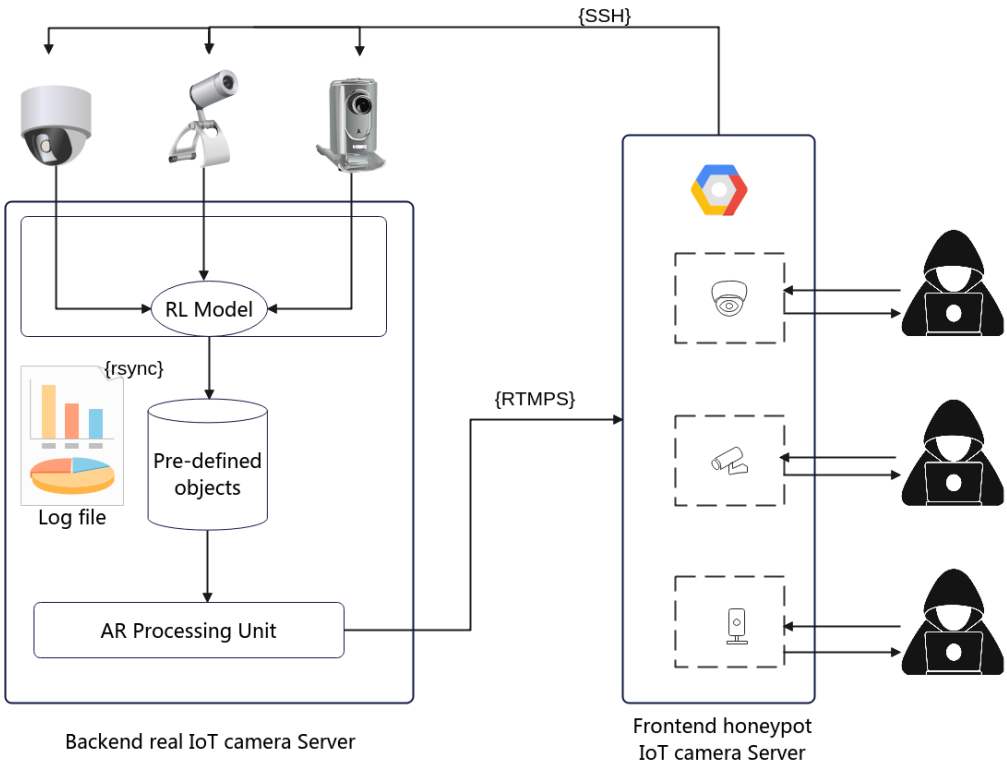


Figure 22 – *General overview of the model.*

Illustrated in Figure 22, our system comprises three fundamental components: *Backend real IoT camera*, *AR processing unit*, and *Frontend honeypot IoT camera*.

The *frontend honeypot IoT camera*, a virtual server, is deliberately placed on the public Internet to attract and interact with potential attackers. Designed to closely mimic the functionality of a real IoT camera, it opens selected ports to offer services such as video streaming, allowing attackers to engage with the camera by issuing control commands like pan, tilt, or zoom.

The role of the *Backend real IoT camera* is to facilitate the live streaming of video content derived directly from real IoT camera sources. This content is transmitted to the *AR processing unit*, synchronized with the respective *frontend honeypot IoT camera*. The *AR processing unit* maintains a direct connection to the Backend, which is grounded in real camera footage and operational camera commands. Consequently, any control commands issued by attackers towards the *frontend honeypot IoT camera* are seamlessly transmitted to the real IoT camera, with corresponding adjustments applied to the field of view based on the attacker's

camera control instructions. This augmented field of view, complete with enhanced objects, is then presented to the attacker.

In the context of the AR experience, it is essential to emphasize that our approach does not grant attackers direct access to manipulate the AR elements, such as adding or removing objects themselves. Instead, attackers are limited to interaction solely through command control. This means that attackers engage with the honeypot by issuing specific commands, such as pan, tilt, or zoom, rather than actively participating in the augmentation process. It is noteworthy to underline that attackers remain unaware of the AR layer within the honeypot environment. Their interaction is channelled exclusively through control commands, and they are not privy to the fact that the environment is enriched with augmented elements. This design decision is intentional, as it contributes to maintaining the deception's authenticity while preventing attackers from discerning the underlying mechanisms employed in our honeypot. The subsequent sections delve into the detailed workings of each of these three components, elaborating on their functionalities and interconnections within our system.

V.3.1.1. Frontend Honeypot IoT Camera

The *frontend honeypot IoT camera* plays a pivotal role in engaging attackers directly, necessitating its behaviour to be indistinguishable from that of an actual IoT camera in the eyes of the attackers. Similar to the approach in [6], by carefully analyzing both commercial IoT cameras and the port scanning tools commonly used by attackers to find vulnerable devices, we can identify the essential services and ports that need to be active in our honeypot for it to appear as a legitimate IoT camera [133]. Below are the key ports our honeypot should open and the services it should mimic:

- **Interactive Web Page (Port 80/8080):** Most IoT cameras provide an interactive web page on ports 80 or 8080, allowing users to view video streams and control the camera.
- **Open Network Video Interface Forum (ONVIF) Support (Port 8000):** IoT cameras adhering to the ONVIF standard expose port 8000 for functions like camera control and real-time video streaming.
- **Real Time Streaming Protocol (RTSP) (Port 554):** RTSP is commonly used on port 554 by IoT cameras to support low-latency video streaming.

- **Real-Time Messaging Protocol (RTMP) (Port 1935):** RTMP, though less common now due to the decline of Adobe Flash Player, is still used by many IoT cameras to stream video over port 1935.

By activating these ports and simulating these services, the *frontend honeypot IoT camera* can effectively engage with attackers. It forwards the control commands to the *backend real IoT camera*, which then generates the corresponding video and instructs the *AR processing unit* to enhance the video stream with the appropriate virtual object. The altered video stream is then sent back to the attacker. The *frontend honeypot IoT camera* is a lightweight server requiring minimal computational resources, making it easy to deploy multiple instances on cloud platforms. This setup ensures smooth interaction with attackers while maintaining the integrity of the deception.

V.3.1.2. Backend Real IoT Camera

At the heart of our system, the *Backend real IoT camera* server plays a multifaceted role that includes video stream distribution to the *AR processing unit*, attacker information recording, and transmission of attacker commands to the *AR processing unit*. Given the pivotal nature of this component, it is imperative to ensure its protection and stealthiness from potential attackers. To this end, we prioritize secure protocols for all communication between the *backend real IoT camera* server and the *frontend honeypots IoT cameras*.

Secure Video Streaming: One of the primary responsibilities of the *backend real IoT camera* server is to securely transmit video streams to the corresponding *frontend honeypots IoT cameras*. To achieve this, we implement the RTMPS protocol, which leverages Secure Sockets Layer (SSL) encryption to establish a secure tunnel for video transmission. RTMPS significantly mitigates the risks associated with cloud-based video streaming. This protocol's adoption is underscored by its extensive use on renowned platforms like YouTube and Facebook Live, reflecting its robustness and security.

Real-Time Secure Communication: Secure communication is paramount between the *backend real IoT camera* server and each *frontend honeypot IoT camera*. To achieve this, Secure Shell (SSH) connections are established to facilitate the real-time transmission of the attacker's camera control commands and attack traces. This ensures that the sensitive information remains encrypted during transit, safeguarding it from potential compromise.

Data Handling and Analysis: Upon receiving attacker commands, the *backend real IoT camera* server orchestrates the *AR processing unit* to augment the

video stream. Simultaneously, it records crucial attack traces, such as connection details and commands, in a dedicated MySQL database. To avert potential server failures and data loss, we implement a backup mechanism. This involves employing the robust and versatile rsync tool, which expedites file transmission and is widely recognized for backup applications.

Additionally, the *backend real IoT camera* server assumes the role of preliminary attack trace analysis. By processing the attack traces, it generates insights that inform subsequent defence strategies. For instance, analyzing the number of attackers interacting with each *frontend IoT camera honeypot* may prompt the deployment of additional honeypots in response to heightened attacker activity, or the removal of less-frequented honeypots.

V.3.1.3. Dynamic Behavior Emulation Through RL

We introduce an RL framework within the backend real IoT camera to enable dynamic adaptation of the honeypot's behaviour based on attacker interactions. The RL environment is created by collecting comprehensive data on real IoT camera behaviours, including responses to various control commands such as zoom, tilt, and pan. Using this data, we train RL algorithms to predict attacker actions and select appropriate augmented objects in real time. For example, if an attacker frequently zooms in on specific areas, the RL agent learns to respond by adding relevant objects to those areas, thereby guiding the attacker's engagement. The Q-learning algorithm is employed to update the Q-values using the Bellman equation, which calculates the expected future reward for a given state-action pair.

Adaptive Responses: With RL in place, the *Backend IoT Camera* server dynamically adapts its responses based on the attacker's actions. *For instance, if an attacker typically zooms in on certain areas, the RL agent learns to respond by choosing objects that match the attacker's interests, effectively guiding the attacker towards engaging areas. These objects are meticulously pre-defined and stored by the defender*. In this section, we delve deeper into the concept of adaptive responses enabled by RL in the *Backend real IoT camera* server.

Data Collection and Learning: The adaptive response process begins with the collection of comprehensive data on real IoT camera behaviours, including how they respond to various control commands, zoom levels, tilt directions and pan motions. This dataset serves as the foundation for training the RL agent.

Creating an RL Environment: An environment is constructed within the backend server that mimics the interactions that an IoT camera honeypot would have with an attacker. This environment offers the RL agent a platform to learn

and make informed decisions based on its actions and the current state of the environment. The Q-learning algorithm [134] adjusts the Q-values using the Bellman equation [135], which computes the expected future reward for each state-action pair. The Bellman equation enables the Q-network to estimate the optimal Q-values by continuously updating them. This is done by calculating the difference between the predicted Q-value and the sum of the immediate reward and the discounted maximum Q-value of the next state-action pair, as shown in

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right], \tag{V.1}$$

where:

- $Q(s_t, a_t)$ represents the Q-value for the state s_t and action a_t;
- α is the learning rate, controlling the magnitude of updates;
- r is the immediate reward received after performing action a_t in state s_t;
- γ is the discount factor, balancing the significance of immediate and future rewards;
- s_{t+1} denotes the subsequent state resulting from action a_t in state s_t;
- a_{t+1} is the action that yields the highest Q-value in the next state s_{t+1}.

In this work, examples of actions and relevant parameters for the RL model include:

- **States** (s): The state of the RL environment can be defined by various attributes of the IoT camera honeypot and the attacker's interactions. Example states include:
  - Camera view angle (pan, tilt, zoom levels).
  - Presence or absence of augmented objects in the video stream.
  - Historical interaction data (e.g., number of previous interactions with the honeypot).
  - Current security settings and system status.
- **Actions** (a): The RL agent can take various actions to manipulate the honeypot's response to the attacker. Example actions include:
  - Adjusting the camera view (e.g., panning left/right, tilting up/down, zooming in/out).

  - Adding or removing augmented objects in the video stream.
  - Changing the security settings (e.g., altering response messages, adjusting system defences).
  - Logging specific interactions or triggering alerts based on attacker behaviour.

- **Rewards (r)**: Rewards are provided to the RL agent based on the effectiveness of its actions in deceiving the attacker and gathering valuable intelligence. Example reward signals include:
  - Positive reward for successful deception (e.g., attacker believes the honeypot is a real device by interacting with the IoT camera).
  - Positive reward for increased interaction time (e.g., attacker spends more time interacting with the honeypot).
  - Negative reward for detection (e.g., attacker realizes it is a honeypot and terminates interaction).

- **Transition probabilities**: The transition probability $P(s_{t+1}|s,a_t)$ defines the likelihood of transitioning to state s_{t+1} given the current state s_t and action a_t. In the context of an IoT honeypot, these probabilities can be influenced by the attacker's response to the honeypot's actions. For example:
  - If the RL agent adjusts the camera view to follow the attacker's commands, the probability of the attacker continuing to interact may increase.
  - If the RL agent adds convincing augmented objects to the video stream, the probability of successful deception may increase.

Training the RL Agent: Algorithm

In the training process, the Q-network aims to minimize the Mean Squared Error (MSE) between the predicted Q-values and the target Q-values. The MSE is defined by

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(Q(s_{t_i},a_{t_i}) - \left[r_i + \gamma \max_{a_{t+1}} Q(s_{t_i+1},a_{t_i+1})\right]\right)^2, \tag{V.2}$$

where:

- N represents the batch size, indicating the number of experiences used for each training iteration;

- s_i is the current state, and a_i is the action taken in that state;
- r_i signifies the immediate reward received after action a_i in state s_i;
- s_i' denotes the subsequent state after taking action a_i in state s_i;
- The equation compares the current Q-value estimate $Q(s_i, a_i)$ with the target Q-value, which is the sum of the immediate reward and the discounted maximum Q-value of the next state.

The MSE guides the network to refine its Q-value estimates to align with the expected rewards over time.

For predicting attacker actions, we employ a prediction loss function defined as

$$\text{Prediction Loss} = \frac{1}{M}\sum_{j=1}^{M}(\hat{a}_j - a_j)^2. \tag{V.3}$$

Here:

- M indicates the number of predicted attacker actions;
- $\hat{a}_j$ represents the j-th predicted attacker action;
- a_j denotes the j-th actual attacker action.

The prediction loss quantifies the discrepancy between the predicted actions and the actual actions of the attacker.

As shown in the Algorithm 3, the RL agent is trained using Deep Q-Networks (DQN) [136]. Through numerous iterations, the agent learns to predict how attackers may behave based on historical data and its interactions with them. As attackers engage with the honeypot, the RL agent observes their behaviour. For instance, if the attacker consistently zooms in on specific regions of the video stream, the RL agent learns to recognize this pattern and predict their future actions. Upon identifying an attacker's recurring behaviour, the RL agent generates adaptive responses. In the case of an attacker who frequently zooms in on particular areas, the agent might learn to choose appropriate objects or content of interest to guide the attacker's attention. These objects are strategically placed in those areas, effectively enticing the attacker to interact further.

The Backend server continuously monitors the attacker's interactions, observing how they engage with the AR elements predicted by the RL agent. The server records data on how long the attacker spends interacting with the augmented elements, the type of objects they engage with, and their responses. As the RL agent

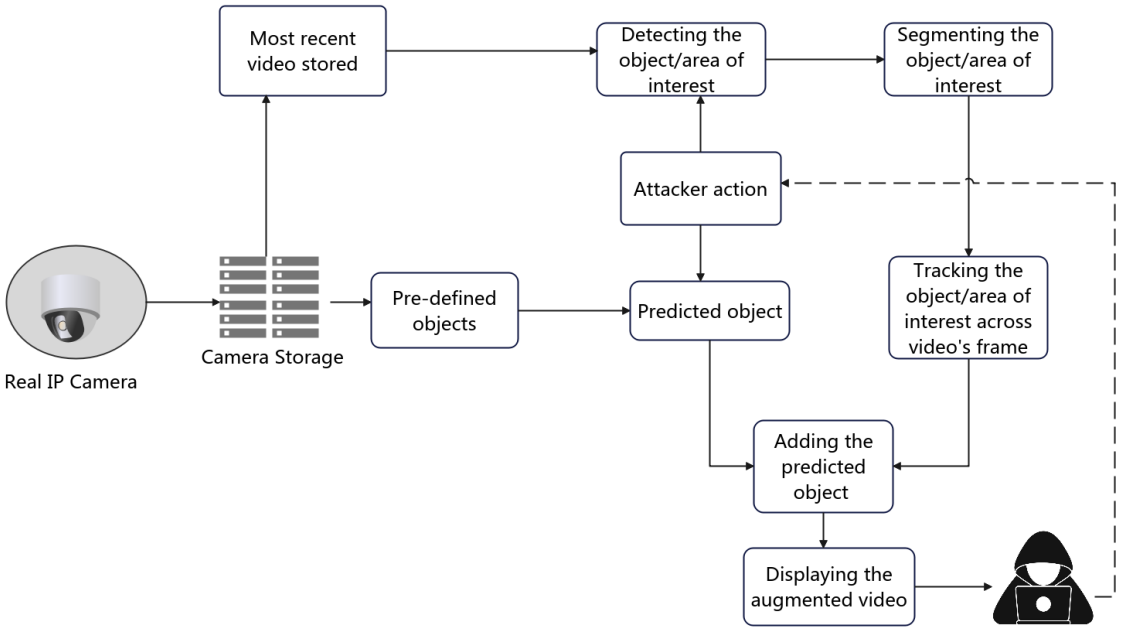


Figure 23 – *AR processing unit steps.*

accumulates more data on attacker interactions, it fine-tunes its responses. This process involves refining the selection, timing, and placement of AR elements to maximize the deception's effectiveness. Over time, the agent's responses become increasingly aligned with attackers' preferences. By effectively guiding the attacker's attention towards engaging areas through adaptive AR responses, the honeypot achieves a higher level of deception. Attackers may be more likely to spend more time interacting with the honeypot, providing researchers with valuable insights into attacker strategies and behaviours.

V.3.1.4. AR Processing Unit

The *AR processing unit* plays a pivotal role in our proposed honeypot framework, enabling us to seamlessly integrate deceptive elements into the video streams extracted from real IoT cameras. Figure 23 depicts the steps of augmenting an object in a video and rendering the augmented video to the attacker. By leveraging AR technology, we elevate the honeypot's capabilities beyond mere video streaming, transforming it into an interactive environment that can engage and mislead potential attackers effectively.

The *AR processing unit* bridges the *backend real IoT camera* and *the frontend honeypot IoT camera*, seamlessly embedding augmented objects into the live video stream. Following is a detailed overview of the components and functionalities of the *AR processing unit*.

```
Input: Replay buffer, Q-network, target network;
Output: Trained Q-network;
Initialization: Initialize Q-network with random weights;
Initialize target network with same weights as Q-network.
for each episode in range(max_episodes) do
    Reset environment,
    get initial state,
    done = False.
    while not done do
        Choose action using epsilon-greedy policy based on Q-network;
        Execute the chosen action in the environment and observe the next state,
          reward, done;
        Store transition in replay buffer;
        if replay buffer size > batch_size then
            Sample random batch from replay buffer.
            for each transition in batch do
                Compute Q-values for the current state using Q-network;
                Compute Q-values for the next state using target network (Double
                  DQN);
                Calculate target Q-value using Bellman equation;
                Update Q-network using mean squared error loss.
                if attacker's behaviour pattern detected then
                    Predict the attacker's next action based on the current state
                      using Q-network;
                    Identify appropriate augmented objects based on predicted
                      action.
                end
            endfor
        end
        if update_target_counter % target_update_frequency == 0 then
            Update target network weights with Q-network weights;
            Update_target_counter = 0.
        end
        Update epsilon for epsilon-greedy policy.
    end
    Print episode results and metrics.
endfor
```

Algorithm 3: Training the RL Agent using DQN.

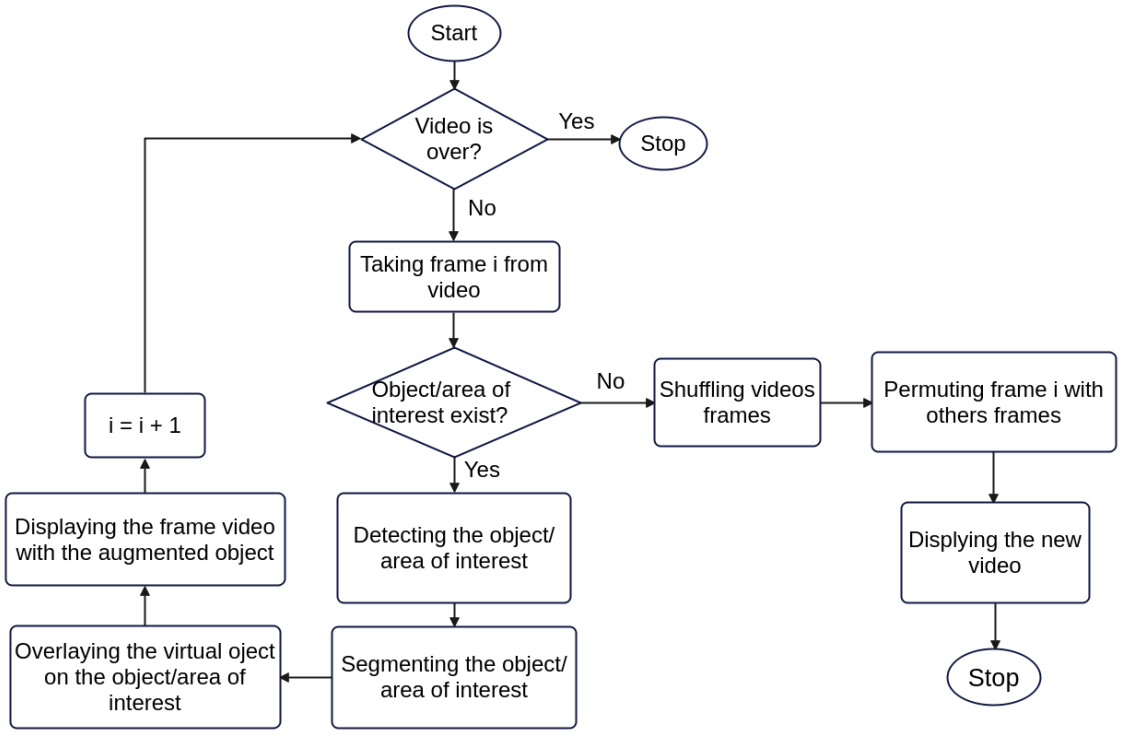


Figure 24 – *Flowchart of the augmented reality processing unit.*

The *AR processing unit* receives the live video stream captured by the *backend real IoT camera*. As presented in Figure 24, it performs real-time video processing to detect and track key features in the video frames, facilitating the accurate placement of augmented objects. This dynamic video processing ensures that the augmented objects align seamlessly with the real-world elements within the video stream, enhancing the deception's realism. Building upon real-time video processing, the *AR processing unit* selectively overlays and removes augmented objects within the live video stream. This process is orchestrated in response to commands received from the attacker through the *frontend honeypot IoT camera*. Augmented objects can be added to divert the attacker's attention or removed to further engage their interest, thus tailoring the honeypot's responses for maximum effectiveness. For high-quality object rendering, accurately estimating the illumination conditions is crucial. Our system employs a shading analysis algorithm, which utilizes the shading cues present in the video frames to analyze the lighting conditions within the real-world video frames. By analyzing the lighting direction, intensity, and colour temperature, we obtain valuable insights into the scene's illumination characteristics. Let $O_{\text{augmented}}$ represent the augmented object and I_{scene} represent the live video frame. We employ shaders that take into account the estimated illumination L_{scene} to render $O_{\text{augmented}}$ with appropriate shading and reflections.

Physically-based rendering (PBR) [137] technique is utilized to simulate the interactions between light and materials, ensuring that the augmented object appears integrated and convincing within the scene.

Object Tracking and Interaction: The *AR processing unit* employs advanced object-tracking algorithms to monitor the movement and positioning of both real-world objects and augmented elements. This tracking ensures that augmented objects remain consistent with the video's perspective and align accurately with the camera's control commands.

The pivotal role of the *AR processing unit* lies in its ability to dynamically track and align both real-world objects and augmented elements within the live video stream. This process demands sophisticated object-tracking algorithms to ensure a seamless connection between these elements, thereby enhancing the overall immersive experience for potential attackers. At the core of our object-tracking approach lies feature detection and matching. We employ advanced computer vision techniques to identify distinctive features within the video frames, allowing us to create reference points for tracking. These features encompass key visual elements such as corners, edges, or textures that are invariant to changes in lighting, scale, or rotation. Let I_t represent the current video frame, and I_{t-1} represent the previous frame. We extract features using the Scale-Invariant Feature Transform (SIFT) algorithm [138]. Feature matching involves establishing correspondences between the features of I_t and I_{t-1}, resulting in a set of matched feature pairs.

To track the motion of objects, we compute the motion vectors between matched feature points in successive frames. These motion vectors provide insights into the displacement and direction of objects' movement. Using the Lucas-Kanade optical flow algorithm[139], we estimate the optical flow vectors for each feature pair, which serve as a representation of object motion. Let (x_t, y_t) be the coordinates of a feature in frame I_t, and (x_{t-1}, y_{t-1}) be the corresponding coordinates in frame I_{t-1}. The optical flow vector (u, v) can be estimated as

$$u = x_t - x_{t-1}, \tag{V.4}$$

$$v = y_t - y_{t-1}. \tag{V.5}$$

Additionally, we leverage homography transformations to rectify the augmented objects' perspective and align them accurately within the video frame. Homography matrices capture the transformation between the real-world coordinates and the augmented objects' projected positions, ensuring a consistent perspective regardless of camera movement.

To enhance tracking stability and accuracy, we integrate Kalman filtering [140] into our tracking algorithm. Kalman filters predict the next object's position based on its previous position and the expected motion, considering uncertainties and noise. By fusing the predicted position with the actual measurements, the Kalman filter offers a refined and stable estimation of object movement. Let x_k represent the state vector that includes position and velocity, and P_k be the error covariance matrix. The prediction step involves updating the state estimate and covariance matrix as given by

$$\hat{x}_k = F_k x_{k-1}, \tag{V.6}$$

$$P_k = F_k P_{k-1} F_k^T + Q_k, \tag{V.7}$$

where F_k is the state transition matrix and Q_k is the process noise covariance matrix.

Through these advanced object tracking techniques, our AR processing unit maintains a cohesive alignment between the real-world elements captured by the live video stream and the augmented objects overlaid on it. This alignment is crucial for creating a convincing and immersive experience for potential attackers, enabling them to interact with the augmented elements seamlessly.

By leveraging RL techniques, the *AR processing unit*'s behaviour is fine-tuned to the attacker's patterns. Importantly, we ensure that the AR experience remains concealed from the attacker's awareness. While the attacker can interact with the *frontend honeypot IoT camera* and perceive the video stream, they are not granted direct control over the *AR processing unit's* actions. This separation ensures that the deceptive elements remain under our control, creating a controlled environment for the attacker's engagement without revealing the true nature of the honeypot.

V.3.2. System Implementation

The implementation of ARPotCam involves deploying augmented reality environments that simulate the behavior of IoT cameras. The system dynamically responds to camera control commands using RL techniques, increasing engagement and deception. The implementation includes the use of 360-degree video streams, real-time interaction capabilities, and advanced monitoring techniques to analyze access patterns and interaction behaviors. The AR processing unit plays a critical role in embedding augmented objects into live video streams captured by the backend real IoT camera, ensuring that the augmented objects align seamlessly with real-world elements. Secure communication protocols and data handling mechanisms are employed to maintain the integrity and confidentiality of interactions between the backend real IoT camera server and frontend honeypot IoT cameras.

V.4. Evaluation and Results

In this section, we evaluate the effectiveness and performance of our innovative AR honeypot system for IoT cameras, shedding light on its capacity to enhance cyber security through deceptive tactics, scalability, and covert operation.

V.4.1. System Implementation

To validate the effectiveness of our proposed approach, we implemented the novel honeypot system and deployed it within the Google Cloud Platform (GCP). This subsection provides a comprehensive step-by-step account of the implementation process, covering the creation of the AR environment, the configuration of the video streams from the backend real IoT camera to the frontend honeypot IoT cameras, and the orchestration of communication and interaction between the system components.

V.4.1.1. Configuring Video Streams

The backbone of our honeypot system lies in the video streams provided by the backend real IoT camera. The real IoT camera captures live video footage, which is then forwarded to the AR processing unit. The frontend honeypot IoT cameras, acting as virtual servers, connect to the AR processing unit to receive the augmented video stream. On the frontend honeypot IoT cameras virtual server, ports were opened to accommodate the various streaming protocols, including port 554 for RTSP and port 1935 for RTMP. Additionally, ports 80 and 8080 were opened to facilitate an interactive web page for camera control, which includes functionalities such as zooming, panning, and tilting.

V.4.1.2. Creation of AR Environment

The AR environment serves as the foundation for our deceptive honeypot system. We developed this environment using a combination of Unity3D and ARCore. By integrating these technologies, we achieved real-time rendering and tracking of augmented objects.

For our experiment, the AR processing unit was hosted on the same machine as the backend IoT camera server to ensure seamless communication. Central to the AR environment's functionality is the repository of augmented objects stored on the backend real IoT camera. These objects are meticulously curated and predefined by the defender. They encompass a diverse range of items, each with its unique attributes and characteristics, contributing to the deceptive potential of the honeypot system.

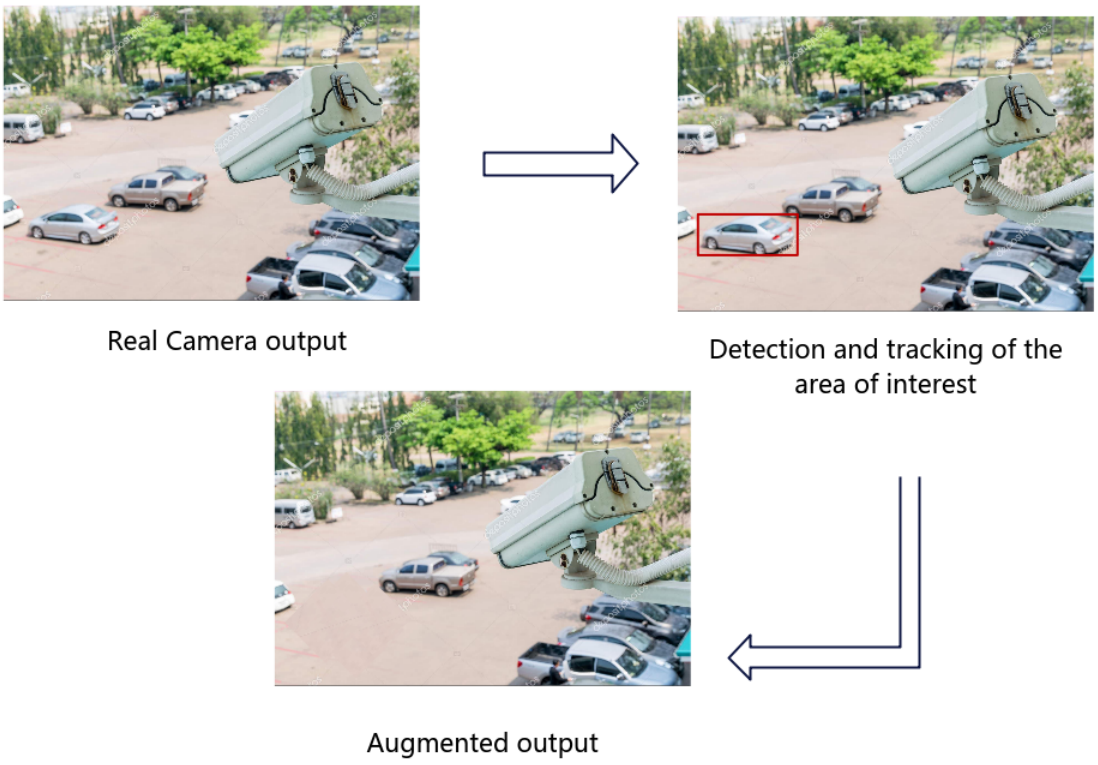


Figure 25 – *Example of AR video frame under the action "remove the car".*

Instead of static object placement, the system dynamically predicts and selects objects in real time, a significant departure from conventional honeypot setups. The choice of an object to be added to the AR environment is not arbitrary but rather a result of sophisticated RL models. This model is trained to predict the attacker's actions and preferences, enabling the system to strategically introduce objects that align with the attacker's interests. *For example, if the attacker consistently zooms in on specific areas, the RL agent can learn to respond by predicting the object to be added based on the attacker's historical actions that match those areas, effectively guiding the attacker's engagement.* This prediction, coupled with the live video stream from the real IoT camera, is then sent to the *AR processing unit*. The unit adds the anticipated augmented object to the video stream using FFmpeg with the x264 encoder. The resulting augmented video is then pushed to the *frontend honeypot IoT cameras*, allowing attackers to interact with the augmented environment. This dynamic and adaptive approach adds a layer of complexity to the honeypot's deception capabilities.

Figure 25 represents an example of an augmented video frame. For our experiment, the AR processing unit was hosted on the same machine as the backend IoT camera server to ensure seamless communication.

Table 7 – *Number of requests received by each honeypot on the GCP web server*

Method	Total number of distinct remote
ARPotCam	**27,869**
HoneyCam [6]	24,120
Real-Camera-Honeypot [5]	26,770
Recorded Video Stream [113]	25,770

V.4.1.3. Communication and Interaction

The communication and interaction between the system components are orchestrated to create a cohesive experience for potential attackers. The *frontend honeypot IoT cameras*, hosted virtually on GCP with public addresses, send attacker control commands to the *backend real IoT camera* server. These commands are relayed through secure SSH connections to maintain confidentiality. The *backend real IoT camera* server not only receives and processes these commands but also records relevant log information, including attacker IP addresses and interaction patterns.

V.4.2. Experimental Result

In this sub-section, we provide a comprehensive evaluation of the effectiveness and performance of our novel honeypot system, focusing on engaging potential attackers and diverting their malicious activities. Analyzing the information from the log file of the *backend real IoT camera* helps us to evaluate the performance of our proposed approach. We outline a set of crucial metrics that we employed for evaluation, present the outcomes of our experiments involving attackers interacting with the honeypot, and provide a detailed comparison with the results obtained from deploying real IoT cameras as honeypots and existing honeypots. The existing works used for comparison in this study, such as those described in [6, 5, 113], were not evaluated by us directly. Instead, we have considered the setup and performance metrics provided by the original authors of those papers

The metrics we selected for evaluating our honeypot's effectiveness were strategically chosen to provide a holistic view of its performance and capabilities.

The number of requests and diversity of attackers: The number of requests is a metric that measures the total number of requests captured by the honeypot. A higher number of requests can indicate a honeypot's attractiveness to attackers, but it does not necessarily mean that useful information was extracted from the attackers. The diversity of attackers is a metric that measures the number of unique attackers that interact with the honeypot. A honeypot that attracts a diverse range

Table 8 – *Types of attack interactions captured*

Method	Port scan attacks	Attackers access the video stream	Attackers interact with camera command
ARPotCam	25,100	2,674	2,012
HoneyCam [6]	24,120	16,08	1,336
Real Camera Honeypot [5]	23,658	1,534	1,032
Recorded Video Stream [113]	23,570	1,034	0

of attackers can provide valuable intelligence about emerging threats and attack trends.

As depicted in Table 7, our novel approach ARPotCam encountered 60,568 connections from 27,869 distinct remote IP addresses spanning the globe throughout 3 months. As shown in Table 8, the majority of these incoming connections appear to be port scans, likely orchestrated by automated scripts. Among these, about 2,674 accessed the video stream, and a mere 2,012 engaged with the camera by transmitting camera control commands. HoneyCam [6] recorded interactions from distinct remote IP addresses globally, with the bulk being attributed to port scans initiated by automated mechanisms. Among these, only 1608 accessed the video stream, and 1336 interacted with the camera using camera control commands. Also, the Real-Camera Honeypot encountered port scans, stemming from 25,770 distinct IP addresses. Among these, 1,534 accessed the video stream from the Real-Camera Honeypot, and 1,032 among them interacted further by issuing camera control commands. On the other hand, the recorded video honeypot faced a similar influx of port scans and video access instances. However, due to the absence of camera control functionalities, no attacker interactions were recorded in this case.

In-depth analysis of port scan statistical distribution unveiled several distinct methods employed by attackers. This analysis was supplemented with an exploration of the subsequent attack patterns after these port scan attempts. Notably as represented, lightweight tools such as cURL and Wget, often used by compromised IoT devices within an IoT botnet, were detected in the context of botnet infection attacks. Precisely, similar to HoneyCam, our proposed honeypot identified the presence of the recent Mozi and Mirai [141] botnet malware across more than 200 distinct IP addresses. Further checking of attackers utilizing port scanning tools [114] like Nmap, Masscan, RustScan, Shodan or IoT-Scan [142] unveiled divergent attack patterns. Specifically, some attackers were able to discern the nature of

Table 9 – *Port scan distribution*

Methodstools	Percentage	Post-Access Actions
cURL / Wget [114]	19.1%	Botnet recruitment/infection
Nmap [114]	8.5%	Access video stream Control camera Directory traversal
IoT-Scan [142]	9.2%	Access video stream Control camera Directory traversal Remote code execution
Shodan [114]	10.3%	Access video stream Control camera Directory traversal Remote code execution
ZMap [114]	6.6%	Access video stream Control camera Directory traversal Remote code execution
Masscan [143]	7.3%	Access video stream Control camera Directory traversal
RustScan [143]	2.1%	Access video stream Control camera
Censys [114]	3.2%	None
Nuclei [114]	1.9%	None
ZGrab [114]	7.7%	Access video stream Control camera
Python request	9.4%	Access video stream Control camera Remote code execution Denial-of-Service (DoS)
Unidentified	14.7%	Access video stream Control camera Directory traversal Botnet recruitment /infection

our ARPotCam as an IoT camera using these tools. Subsequently, these attackers accessed the video stream and exercised camera control commands via the interactive webpage of our *frontend honeypot IoT camera*. Additionally, they initiated directory traversal attacks targeting the web server associated with our *frontend honeypot IoT camera*.

Engagement Time: This metric captures the duration an attacker spends interacting with the honeypot. It serves as an indicator of the system's ability to

maintain attacker interest over an extended period, enhancing the potential for data collection and deception. As shown in Figure 26, the attacker interacts more with the proposed honeypot. The height of the bars represents the number of interactions falling within that time range.

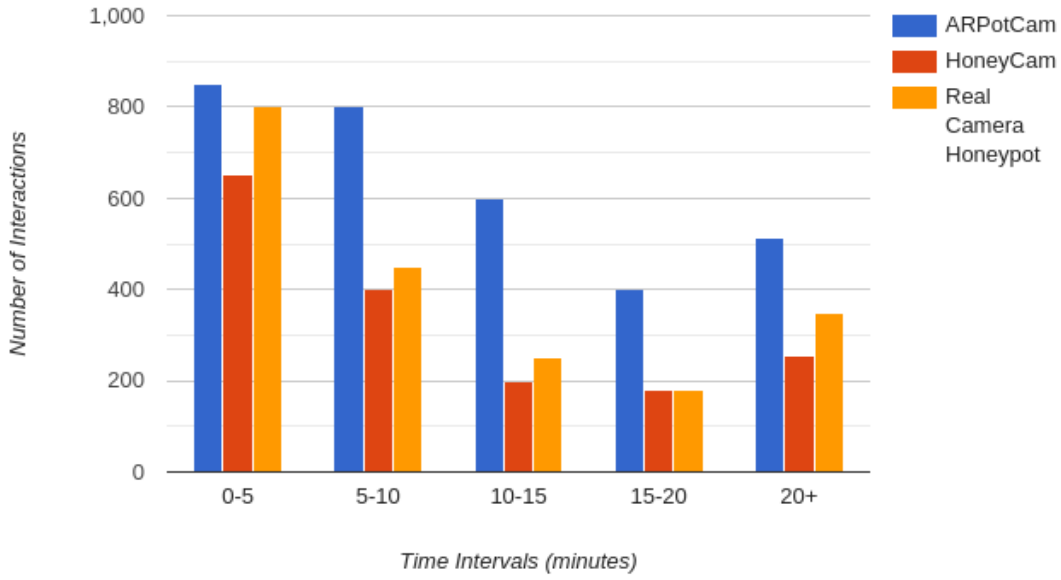


Figure 26 – *Duration an attacker spends interacting with the honeypot.*

Commands Executed: The number of camera control commands issued by the attacker illustrates the level of engagement with the honeypot's functionalities. A higher number indicates effective deception and a responsive system. As shown in Table 9, attackers send some commands to infect or recruit our honeypot in the botnet. We also observe commands to access the video stream where some attackers go forward and interact with camera control. Additionally, they initiated directory traversal attacks targeting the web server associated with our honeypot. In Figure 27, we see that ARPotCam has a higher number of commands executed over time compared to HoneyCam and Real-Camera Honeypot.

Figure 28 provides an overview of the camera control commands, specifically zooming, panning left/right, and tilt up/down, received by ARPotCam, HoneyCam and Real-camera Honeypot. In the case of ARPotCam, approximately 54.3% of the commands involve zooming in or out, while 28.9% entails tilting up or down, and the remaining 16.8% relates to panning left or right. These distributions are similar to those observed for Real-Camera Honeypot, where 52.7% of the com-

mands involve zooming in or out, 33.8% entail tilting up or down, and 13.5% are associated with panning left or right.

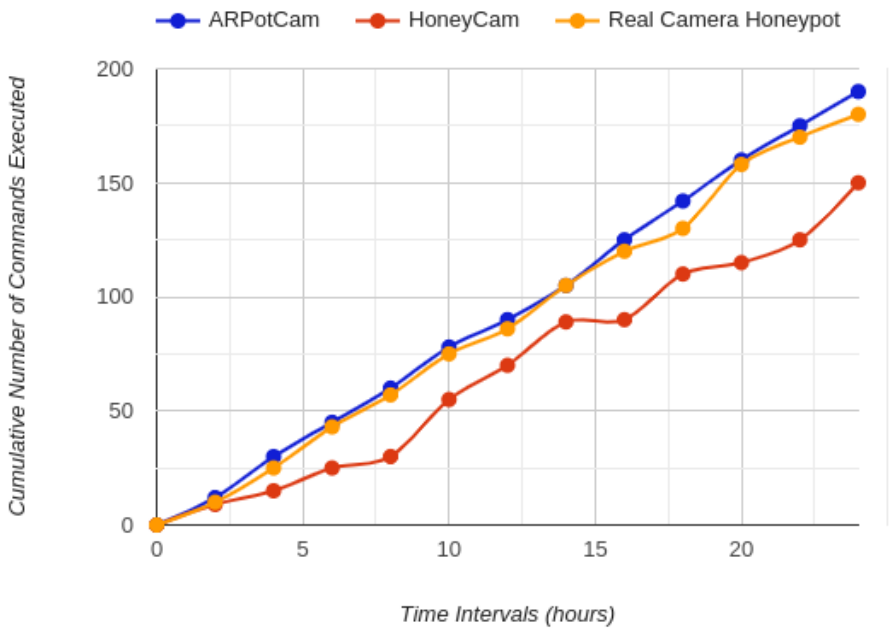


Figure 27 – *Number of camera control commands issued by the attacker.*

Deception Rate: The proportion of attackers misled by the honeypot is critical to gauge its efficacy. This metric reflects the success of the AR environment in convincingly replicating authentic IoT camera interactions. The deception rate was measured by analyzing the behaviour and interaction patterns of the attackers with the honeypot. Specifically, an attacker was considered deceived if they spent a significant amount of time interacting with the honeypot, issued multiple camera control commands, or accessed the video stream repeatedly without detecting the honeypot's true nature. In contrast, attackers who quickly disengaged or did not perform any meaningful interactions after an initial probe was considered not deceived. Figure 29 shows that approximately 72.5% of attackers were successfully deceived, while about 27.5% did not extend their interaction with the honeypot.

Attack Traces: Collecting logs of attacker actions, IPs, and interaction patterns enables a post-analysis of their behaviour. This data assists in refining the honeypot's strategies and enhancing its deception capabilities. In Table 9 we see different follow-up attacks made by attackers after the port scan.

Adaptive Responses: The frequency and success of adaptive responses generated by the RL agent reflect the system's agility in tailoring deception tactics. This

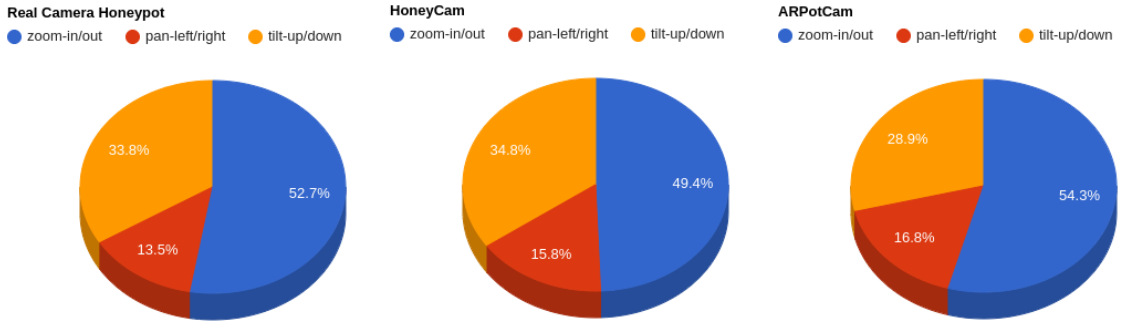


Figure 28 – *Varieties of camera control command distribution. (a) Real-Camera Honeypot [5], (b) HoneyCam [6], (c) ARPotCam .*

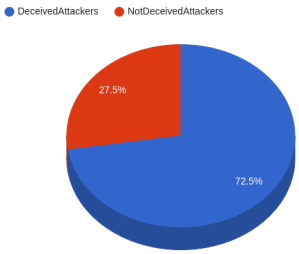


Figure 29 – *Proportion of attackers misled by the honeypot.*

metric demonstrates the honeypot's capacity to dynamically adjust its responses based on attacker actions.

Figure 30 provides a comprehensive view of the performance of ARPotCam over 3 months experimental duration. The first graph, displaying the frequency of successful adaptive responses over time, highlights how often our system effectively tailored deception tactics. It offers insights into trends and fluctuations in our system's performance. This graph allows us to pinpoint that during week 5, our system excelled and during week 4 faced challenges in responding to attackers. The second graph, showcasing the success rate of adaptive responses over time, offers a more quantitative perspective. It calculates the proportion of successful responses out of total responses. The analysis of this graph reveals that our system's adaptive responses improved throughout the 6 weeks. Finally, the third graph, illustrating cumulative successful adaptive responses over time, tracks the overall effectiveness of our system. It shows how the total count of successful responses increased week by week. Together, these graphs enable us to thoroughly evaluate the agility and effectiveness of our system in tailoring deception tactics over time. They help us identify areas for improvement and better understand how our system responds to attackers' actions and commands throughout the experiment.

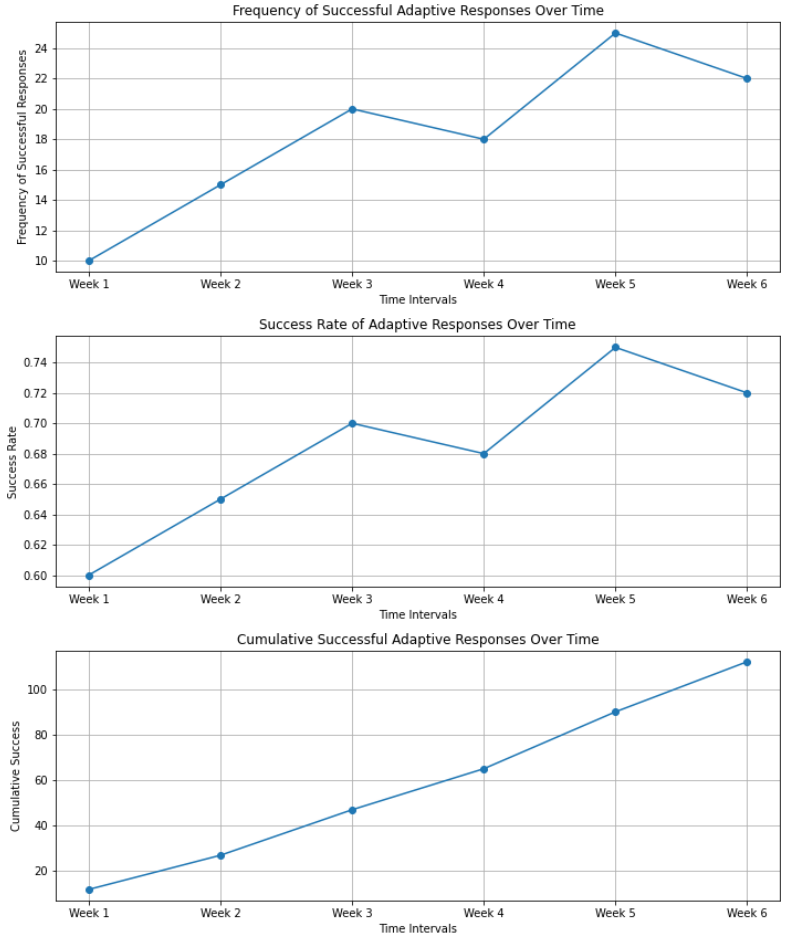


Figure 30 – *Frequency and success of adaptive responses generated by the RL.*

Scalability: Measuring the honeypot's ability to handle multiple concurrent interactions while maintaining performance and realism is essential. Scalability ensures that the honeypot remains effective even under increased load. Our virtual honeypot demonstrated superior scalability compared to real IoT camera honeypots. The virtual nature of our setup enabled it to handle more concurrent interactions without compromising performance.

To support our claims on scalability, we conducted experiments where the honeypot was subjected to varying levels of concurrent interactions. The results, shown in Table 10, indicate that our virtual honeypot maintained high performance and responsiveness, even with an increased number of simultaneous attackers. The honeypot successfully handled up to 500 concurrent interactions, maintaining a stable response time and low error rates, demonstrating its capability to scale effectively.

Covertness: Assessing the degree of attacker awareness regarding their interaction with a honeypot is crucial. A higher level of covertness indicates the honeypot's success in concealing its deceptive nature. Incorporating AR significantly

Table 10 – *Scalability performance metrics*

Number of concurrent interactions	Average response time (ms)	Error Rate (%)
50	200	0.5
100	210	0.7
200	220	0.9
500	250	1.2

enhanced the honeypot's covertness. Attackers found it challenging to differentiate between the tricky environment and a real IoT camera.

To assess the covertness of ARPotCam, we examined its resistance to detection by commonly used reconnaissance and honeypot identification tools. One of the most widely recognized tools for this purpose is the Shodan honeyscore [144], which estimates the likelihood of a device being a honeypot. The Shodan honeyscore assigns a value between 0.0 (definitely a real system) and 1.0 (definitely a honeypot) to an IP address. This score is influenced by various factors such as the number of open ports, known honeypot fingerprints, interaction levels with the host, and the history of the IP address. The score is computed using a proprietary machine learning classification algorithm. Our objective was to evaluate how effectively ARPotCam could evade this advanced reconnaissance tool.

After deploying ARPotCam, we monitored it for a week before utilizing the Shodan API to retrieve the honeyscores of the honeypots by providing their respective IP addresses. As shown in Figure 31, ARPotCam exhibited a covertness level similar to that of the real-camera-based honeypot, both achieving a honeyscore of 0.01. In contrast, other honeypots demonstrated significantly lower levels of covertness. This high level of covertness demonstrates the effectiveness of our AR-enhanced environment in deceiving attackers.

Timing and Synchronization: One notable concern is the delay between the real video stream captured by the IoT camera and the augmented video stream transmitted to potential attackers. Delays can be introduced during the capture, processing, and rendering stages, as well as in the transmission itself, depending on network conditions and protocols. Achieving low latency between the real and augmented content is paramount to maintaining a convincing illusion for attackers and enhancing the honeypot's effectiveness.

To address this, we measured the end-to-end delay from video capture to the augmented stream received by attackers. The results, presented in Table 11, show that our system achieved an average delay of 150ms, with a maximum delay of 200ms under peak load conditions. These measurements indicate that our system

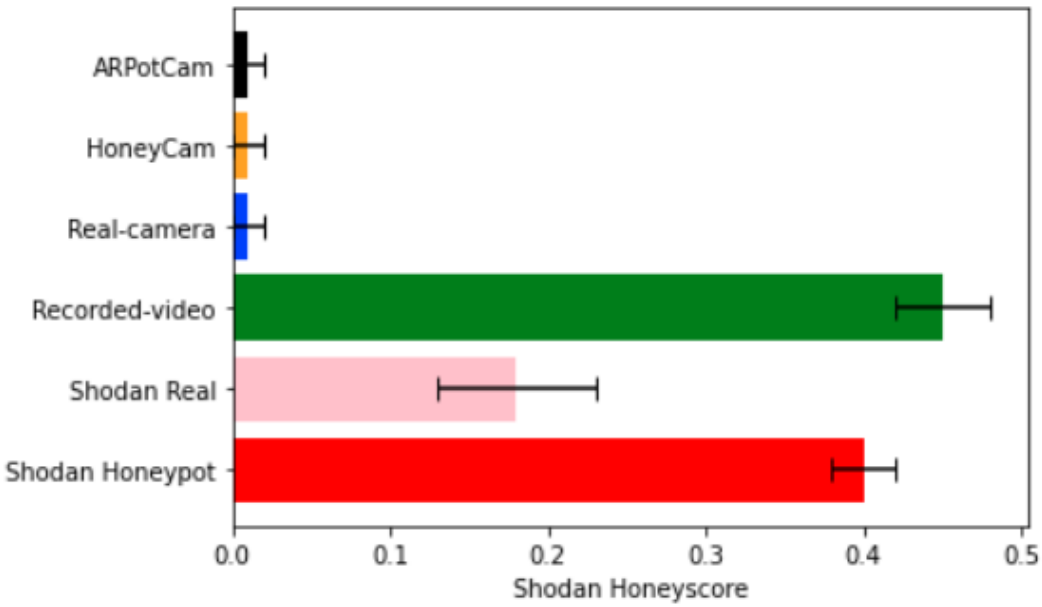


Figure 31 – *Covertness Evaluation: Percentage of Attackers Deceived*

Table 11 – *Delay measurements for augmented video stream*

Condition	Average Delay (ms)
Normal Load	150
Peak Load	200

is capable of maintaining low latency, ensuring a seamless and realistic interaction for attackers.

V.5. Analysis and Discussions

V.5.1. Analysis

The evaluation of the ARPotCam system highlights its robust capabilities in improving IoT camera security through innovative deception tactics. Our experiments, conducted over three months, revealed several key insights into the system's performance, engagement, and adaptability.

The number of requests and diversity of attackers served as primary indicators of the honeypot's effectiveness. ARPotCam encountered a substantial number of interactions, 27,869 connections from distinct remote IP addresses. This high level of engagement underscores the honeypot's attractiveness to attackers and its ability to draw diverse threat actors. Notably, the system recorded a significant number of interactions involving video stream access and camera control commands, fur-

ther validating its effectiveness in engaging attackers. Engagement time, a crucial metric, demonstrated ARPotCam's capability to maintain attacker interest over extended periods. The honeypot successfully engaged attackers for longer durations compared to traditional methods, as illustrated in Figure 26. This extended engagement increases the likelihood of gathering valuable intelligence and understanding attacker behaviors. The number of camera control commands executed by attackers also highlights the honeypot's interactivity. ARPotCam recorded a higher number of commands executed over time, indicating its effectiveness in providing a realistic and responsive environment for attackers. The distribution of commands, including zoom, tilt, and pan, further emphasizes the system's capability to mimic real IoT camera interactions convincingly.

The deception rate, a critical measure of the honeypot's efficacy, revealed that approximately 72.5% of attackers were successfully misled by ARPotCam. This high deception rate underscores the effectiveness of integrating AR and RL techniques in creating a convincing and immersive honeypot environment. The ability to adapt and tailor deception tactics based on real-time attacker interactions significantly contributed to this success. The collection of attack traces, including logs of attacker actions and interaction patterns, provided valuable insights for refining the honeypot's strategies. The adaptive responses generated by the RL agent, as shown in Figure 30, demonstrated the system's agility in tailoring deception tactics to attacker behaviors. This dynamic adaptability enhances the honeypot's effectiveness in misleading attackers and gathering actionable intelligence. Scalability, another important aspect, was validated through experiments demonstrating ARPotCam's ability to handle increased load without performance degradation. The virtual nature of the honeypot setup enabled it to manage multiple concurrent interactions effectively, maintaining high performance and responsiveness, as shown in Table 10.

The covertness of ARPotCam was also a significant achievement, with the system demonstrating a high level of concealment from attackers. The Shodan honeyscore evaluation revealed that ARPotCam maintained the same level of covertness as a real-camera-based honeypot, with a score of 0.01. This result highlights the effectiveness of AR integration in deceiving attackers and concealing the honeypot's true nature. Timing and synchronization between the real and augmented video streams were critical for maintaining a convincing illusion for attackers. Our measurements indicated an average delay of 150ms, with a maximum delay of 200ms under peak load conditions. These low latency figures ensure a seamless and realistic interaction for attackers, enhancing the honeypot's effectiveness.

Overall, the originality of ARPotCam lies in its innovative combination of AR and RL, creating a dynamic and highly interactive honeypot environment. This originality not only sets it apart from traditional methods but also demonstrates its potential for broader applicability. The system's design allows for re-utilization in various IoT contexts, making it a versatile solution for improving security across different types of IoT devices. The adaptability and responsiveness of ARPotCam ensure that it can be easily be used to meet specific security needs, providing a reusable and effective tool for ongoing security in IoT.

V.5.2. Discussions

The ARPotCam system presents several key differences and advantages over existing honeypot approaches for IoT cameras. Traditional honeypots often focus on network interactions and may lack the capability to provide a highly interactive and immersive environment. In contrast, ARPotCam leverages augmented reality and reinforcement learning to create a more convincing and engaging honeypot experience.

One of the main advantages of ARPotCam is its ability to dynamically adapt its deception tactics based on real-time attacker interactions. Traditional honeypots typically rely on static configurations, which can be easily detected by sophisticated attackers. The use of RL in ARPotCam allows the system to learn from attacker behaviors and adjust its responses accordingly, maintaining a high level of deception and engagement.

The integration of AR technology also enhances the realism of the honeypot environment. By overlaying augmented objects onto the live video stream from real IoT cameras, ARPotCam creates a seamless and interactive experience that closely mimics real-world interactions. This level of realism is difficult to achieve with traditional honeypots, which often rely on pre-recorded or static content.

However, there are also some challenges and limitations to consider. One notable challenge is the computational complexity and resource requirements associated with real-time AR processing and RL. Ensuring low latency and high performance in a scalable manner requires careful optimization and resource management. Additionally, maintaining the covertness of the honeypot while integrating advanced AR features poses a significant challenge, as attackers may become increasingly sophisticated in detecting honeypot environments.

The evaluation results indicate that ARPotCam successfully addresses these challenges, demonstrating superior performance in key metrics such as engagement

time, deception rate, and adaptability. The system's ability to handle increased load without performance degradation underscores its scalability and efficiency.

In comparison to HoneyCam, RealCamera Honeypot and the record video for Honeypot, ARPotCam demonstrates significant improvements in engaging and deceiving attackers. The system's dynamic adaptability and realistic AR environment offer a more effective solution for IoT camera security, providing robust protection against cyber threats.

V.6. Conclusion

In this chapter, we explore the convergence of IoT cameras, AR, and honeypots to develop a novel approach called **ARPotCam**. This innovative honeypot system leverages AR and RL to create a deceptive environment that effectively emulates the behavior of IoT cameras, thereby enhancing cybersecurity. By integrating augmented objects into live video streams and dynamically responding to attacker control commands, ARPotCam presents a convincing and interactive decoy that successfully engages and misleads attackers.

Our experimental evaluation demonstrated the effectiveness of ARPotCam in several key metrics, including the number of attacker interactions, engagement time, and deception rate. The system's ability to handle multiple concurrent interactions without compromising performance highlighted its scalability. Additionally, the high level of covertness achieved against state-of-the-art reconnaissance tools underscored the honeypot's capability to conceal its true nature. The results showed that ARPotCam outperforms traditional honeypot approaches by providing a more immersive and realistic experience for attackers, thereby capturing a broader spectrum of attacker behaviors and strategies. This enhanced engagement allows for the collection of valuable threat intelligence, contributing to the development of more robust and adaptive cybersecurity measures.

General Conclusion

CONTENTS

Problem Studies 128
Summary of Research Findings 128
Limitations of the Study 130
Future Research Directions 131

Problem Studies

Throughout this thesis, we have explored the multifaceted challenges of securing IoT devices against sophisticated cyber threats. The exponential growth in the number of interconnected devices, ranging from household appliances to critical industrial equipment, has exposed significant security vulnerabilities. These vulnerabilities are primarily driven by the diverse and resource-constrained nature of IoT devices, which often lack the computational power, memory, and energy resources required for robust security implementations. Furthermore, rapid development and deployment of IoT devices often prioritize functionality and market delivery over security, resulting in widespread weaknesses such as hardcoded passwords, lack of encryption, and limited user configurability creating ripe opportunities for cyber exploitation.

Summary of Research Findings

To address these vulnerabilities, we adopted a comprehensive and proactive approach that integrates ML and cyber deception techniques. Our methodologies were designed to improve the security posture of IoT ecosystems by anticipating, detecting, and mitigating potential threats before they could cause significant

harm. Our research has yielded several significant findings that demonstrate the inadequacy of traditional security measures in protecting IoT devices from advanced cyber threats and the efficacy of innovative approaches integrating machine learning and cyber deception. Although our methodologies have demonstrated potential in mitigating common security issues, this section critically examines their effectiveness and the inherent limitations present in such innovative solutions.

Our first major methodology involved the development of adaptive intelligent interaction honeypots. These honeypots were designed to engage and mislead attackers during the reconnaissance phase of their attacks. By simulating vulnerable IoT devices and dynamically adapting to attacker behaviors, these honeypots effectively captured valuable data on attack methods and tactics. This data was then used to improve the honeypot's deception capabilities, making it more challenging for attackers to distinguish between real and decoy devices. While this contribution has been effective in gathering intelligence on attacker methods and tactics, the approach assumes a static attacker behavior model that may not adapt dynamically to more sophisticated or evolving cyber threats. This limitation suggests that while the honeypots can mislead less advanced attackers, they might fall short against more complex or adaptive threats, thereby questioning their efficacy in continuously evolving security landscapes.

In addition to honeypots, we developed a hybrid recommender system that leverages adaptive learning to propose fake vulnerabilities alongside real ones. This system aims to deceive attackers that might bypass the initial honeypot traps or those originating within the network. By presenting a mix of real and deceptive vulnerabilities, the recommender system confuses and delays attackers, reducing the likelihood of successful breaches. The adaptive learning component ensures that the system evolves in response to changing attack patterns, maintaining its effectiveness over time. This system, though innovative, faces challenges in scalability and real-time adaptability. The critical reliance on continuous updates and the need to respond to evolving attacker strategies without human intervention highlight a significant gap in current ML applications within cybersecurity. The effectiveness of this system in real-world scenarios remains contingent on its ability to adapt faster than the attackers, a challenging feat given the rapid advancement of attack technologies.

Another innovative approach we explored was the implementation of AR-based deception strategies, particularly for IoT cameras. These AR honeypots create realistic and interactive decoys that engage attackers for extended periods. By providing a visually convincing environment, these honeypots can gather detailed intel-

ligence on attacker strategies and tools. The use of AR enhances the believability of the deception, making it more likely that attackers will be fooled and remain engaged. These AR honeypots, while providing a visually immersive environment to engage attackers, require significant computational resources and may introduce latency issues that could tip off experienced attackers. The realism provided by AR is commendable; however, the potential for detection due to technological limitations or savvy attackers could undermine the long-term viability of this strategy.

Limitations of the Study

This research, despite its innovative approach integrating ML and cyber deception to enhance IoT security, encounters limitations that could affect the generalizability and practical implementation of the findings. A significant limitation is the reliance on contemporary technological frameworks and tools that may quickly become obsolete. Although this study has made strides in adapting to evolving threats through dynamic and intelligent deception techniques, the rapid evolution of IoT devices and the emergence of new cyber threats pose continuous challenges. Keeping up with these advancements requires ongoing development and updates to the security measures proposed, which may not always keep speed with the speed of technological change.

Moreover, while the adaptive honeypots and augmented reality systems demonstrated effectiveness in controlled environments, scaling these solutions across a broader network introduces substantial challenges. The resource demands for deploying these systems extensively can be prohibitive, especially in larger, more complex IoT environments. Managing these resources efficiently while maintaining the adaptability and responsiveness of the security measures is an area that needs further research and development. The complexity of orchestrating a large-scale deployment of these advanced systems raises concerns about their practicality and sustainability in everyday IoT applications.

Additionally, the use of cyber deception raises ethical and legal concerns that were only partially addressed in this study. The balance between effective deception and ethical deployment remains delicate, requiring careful consideration of the implications of misleading potential attackers. While the research has established a foundation for the responsible use of deception techniques, comprehensive guidelines and legal frameworks are still needed to ensure these methods do not violate privacy rights or legal standards. Future research should focus on establishing clear

ethical guidelines and ensuring that the deployment of cyber deception techniques adheres to both national and international laws.

Future Research Directions

While our research has made significant strides in enhancing IoT security, several areas deserve further investigation. The limitations identified in this thesis highlight specific directions for future work, which can build on the foundations applied here to advance the field further.

- **Advanced Machine Learning for IoT Security:** Future research should explore the integration of advanced ML models, such as deep learning and ensemble methods, to improve the accuracy and adaptability of security systems. Deep learning models can handle large datasets and complex patterns, enhancing detection capabilities. Ensemble methods, which combine multiple learning algorithms, can improve prediction performance and robustness. Incorporating these models into our current framework could lead to more effective and resilient IoT security solutions.
- **Development of Comprehensive IoT Security Datasets:** There is a need to develop more comprehensive datasets that capture a wider range of IoT device behaviors and attack vectors. The datasets used in this thesis were limited in scope, focusing on specific devices and attack scenarios. Expanding the datasets to include diverse devices and a broader spectrum of attack techniques will better train and evaluate machine learning models. This effort will ensure the security solutions are applicable to a wide range of real-world IoT environments, enhancing their generalizability and effectiveness.
- **Scalability of IoT Security Frameworks:** Future research should focus on the scalability of the security systems developed in this thesis. While our methodologies were effective in controlled environments, deploying these solutions in large-scale IoT ecosystems presents additional challenges. Investigating how the proposed security frameworks can be scaled to handle millions of interconnected devices without compromising performance is crucial. This includes optimizing computational resources, managing large volumes of data, and ensuring real-time threat detection and response capabilities.

- **Ethical and Privacy Considerations in Cyber Deception:** The ethical implications and potential privacy concerns associated with the deployment of cyber deception techniques in IoT security should be thoroughly investigated. While cyber deception can be an effective defense mechanism, it raises ethical questions about the intentional misleading of attackers and the potential misuse of deceptive data. Future research should develop guidelines and frameworks to ensure that cyber deception is used responsibly, balancing security needs with ethical considerations. This includes studying the legal implications and establishing best practices for the deployment of such techniques.

- **Long-Term Sustainability of Adaptive Security Systems:** It is essential to explore the long-term impact and sustainability of adaptive security systems. Our research demonstrated the short-term effectiveness of adaptive honeypots and recommender systems, but their long-term efficacy remains uncertain. Future work should investigate how these systems perform over extended periods and how they can be maintained and updated to remain effective against evolving threats. This includes developing mechanisms for continuous learning and adaptation, ensuring that the security solutions can keep pace with the rapidly changing threat landscape.

  Expanding the Scope of Augmented Reality Honeypots: While this research has focused on using AR honeypots for IoT cameras, future work could explore expanding this approach to a broader range of IoT applications. Many IoT devices, including smart home assistants, industrial sensors, and medical IoT systems, could benefit from deceptive AR techniques to mislead attackers and enhance security. Investigating how AR-based deception can be adapted to different IoT environments would provide valuable insights into its scalability and effectiveness in diverse real-world scenarios.

Bibliography

[1] Alqarawi G, Alkhalifah B, Alharbi N, El Khediri S. Internet-of-things security and vulnerabilities: case study. Journal of Applied Security Research. 2023;18(3):559-75. Available from: https://doi.org/10.1080/19361610.2022.2031841.

[2] Bulut C, Wu PF. More than two decades of research on IoT in agriculture: a systematic literature review. Internet Research. 2023;34:994-1016. Available from: https://doi.org/10.1047/975-3-19-41087-30.

[3] Zhu M, Anwar AH, Wan Z, Cho JH, Kamhoua CA, Singh MP. A survey of defensive deception: Approaches using game theory and machine learning. IEEE Communications Surveys & Tutorials. 2021;23(4):2460-93. Available from: https://doi.org/10.1109/COMST.2021.3102874.

[4] Gayatri T, Srinivasu G, Chaitanya D, Sharma V. A Review on Optimization Techniques of Antennas Using AI and ML/DL Algorithms. International Journal of Advances in Microwave Technology. 2022;7(2):288-95. Available from: https://doi.org/10.32452/IJAMT.2022.288295.

[5] Wang P, Li X. IoTCMal: Authentic IoT Cameras for Deception and Engagement. Computers & Security. 2020;95:101868. Available from: https://doi.org/10.1016/j.cose.2020.101868.

[6] Guan Y, Ma L. HoneyCam: Replicating IoT Camera Functionalities with Pre-Recorded 360° Video Content. Journal of Information Security and Applications. 2022;66:102941. Available from: https://doi.org/10.1016/j.jisa.2022.102941.

[7] Tundalwar DS, Pandhare R, Digalwar M. A Taxonomy of IoT Security Attacks and Emerging Solutions. In: 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS); 2023. p. 1-5. Available from: https://doi.org/10.1009/908-3-319-4117-3.

[8] Kaur G, Kaur H. A Comprehensive Review of Honeypot Research Projects. Journal of Information Security and Applications. 2018;41:102-18. Available from: https://doi.org/10.1016/j.jisa.2018.04.005.

[9] Ashton K. That 'Internet of Things' Thing; 1999. Accessed: 2024-06-19. Available from: https://www.rfidjournal.com/that-internet-of-things-thing.

[10] Cambridge Dictionary. Internet of Things; 2024. Accessed: 2024-06-19. Available from: https://dictionary.cambridge.org/dictionary/english/internet-of-things.

[11] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. In: Future Generation Computer Systems. vol. 29. Elsevier; 2013. p. 1645-60. Available from: https://doi.org/10.1016/j.future.2013.01.010.

[12] Parvathy K, Nataraj B. An Overview on IoT Architecture Application Layer and Security Threats. In: 2023 Third International Conference on Smart Technologies, Communication and Robotics (STCR). vol. 1; 2023. p. 1-4. Available from: https://doi.org/10.1407/978-3-319-d83304d-3.

[13] Alam T. Design a blockchain-based middleware layer in the Internet of Things Architecture. JOIV : International Journal on Informatics Visualization. 2020. Available from: https://doi.org/10.1005/968-3-019-41087-3.

[14] Monges Y, Lopez M, Flecha E, Chaparro E. Performance Evaluation of MQTT and CoAP Communication Protocols in IIoT for Electrical Substations Based on the IEC 61850 Standard. In: IEEE International Conference on Industry Applications; 2023. p. 774-9. Available from: https://doi.org/19.1009/978-53-39-41087-2.

[15] J C, Kulkarni RD, Alhawsawi E, Majid MA. High-Performance and Energy-Efficient FIR Filter Architecture Using Parallel Prefix Adder-Based Triangular Common Subexpression Elimination Algorithm for IoT Enabled Wireless Sensor Network. In: 2023 3rd Asian Conference on Innovation in Technology (ASIANCON); 2023. p. 1-8. Available from: https://doi.org/10.1887/8778-3-319-89787-43.

[16] Romanov A. Smart Home technology: automation system and industry development opportunities. Upravlenie kachestvom (Quality management). 2023. Available from: https://doi.org/10.1957/98-3-19-1187-5.

[17] Zeidan F, ElHayani M, Soubra H. RT DL Tasks Distribution for Sensitive Data Protection and Resource Optimization. In: International Conference on the Internet, Cyber Security and Information Systems; 2023. p. 276-82. Available from: https://doi.org/18.1007/978-3-309-187-0.

[18] Stallings W, Brown L. Computer Security: Principles and Practice. Pearson; 2016. Available from: https://doi.org/978-0134550853.

[19] Weber RH. Internet of Things – New security and privacy challenges. Computer Law & Security Review. 2010;26(1):23-30. Available from: https://doi.org/10.1016/j.clsr.2009.11.008.

[20] Roman R, Zhou J, Lopez J. On the features and challenges of security and privacy in distributed internet of things. Computer Networks. 2013;57(10):2266-79. Available from: https://doi.org/10.1016/j.comnet.2012.12.018.

[21] Sicari S, Rizzardi A, Grieco LA, Coen-Porisini A. Security, privacy and trust in Internet of Things: The road ahead. Computer Networks. 2015;76:146-64. Available from: https://doi.org/10.1016/j.comnet.2014.11.008.

[22] Zhang R, Liu L, Chen Y, Xiang Y. Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control. IEEE Internet of Things Journal. 2014;4(5):1474-84. Available from: https://doi.org/10.1109/JIOT.2014.2351977.

[23] Neupane R, Zobrist T, Neupane K, Bedford S, Prabhudev S, Haughton T, et al. CICADA: Cloud-based Intelligent Classification and Active Defense Approach for IoT Security. IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). 2023:1-6. Available from: https://doi.org/10.1007/978-893-319-41187-3.

[24] Grant TD. Detect, Deny, Degrade, Disrupt, Destroy, Deceive: which is the greatest in OCO? European Conference on Cyber Warfare and Security. 2023. Available from: https://doi.org/10.34190/ew.23.022.

[25] Rowe NC, Rrushi J. Introduction to Cyberdeception. Springer; 2016. Available from: https://doi.org/10.1007/978-3-319-41187-3.

[26] Warren EJ. All Things Wise and Wonderful: A Christian Understanding of How and Why Things Happen, in Light of COVID-19. Perspectives on Science and Christian Faith. 2021. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/4fe7733015d15c5f191c1882fae566e5a410904b.

[27] Yoon S, Jeong I, Kim JJ, Kang B. CORRELATES OF MILD BEHAVIORAL IMPAIRMENT IN OLDER ADULTS WITH COGNITIVE DECLINE. Innovation in Aging. 2023;7:1010 1011. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/84a3c989ff09063b8bc3be8b6b5c2304cb08b557.

[28] Sunarso B, Tusriyanto, Mustafa F. Analysing the Role of Visual Content in Increasing Attraction and Conversion in MSME Digital Marketing. Journal of Contemporary Administration and Management (ADMAN). 2023. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/80a5bbac3f626a445257337bf752b2bb0352fc3c.

[29] Caso L, Cavagnis L, Vrij A, Palena N. Cues to deception: can complications, common knowledge details, and self-handicapping strategies discriminate between truths, embedded lies and outright lies in an Italian-speaking sample? Frontiers in Psychology. 2023;14. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/638b36b6535e7e6dd1cbf465c21e276d292d2b29.

[30] Şen A. Political Public Relations and Crisis Communication of the Government: The Case of the Earthquake in Turkey. Journal of Ekonomi. 2023. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/24659d5f57a6b2d97174239d66e0450e1c4b6ff4.

[31] Çakmak S, Kazan F. A SECTIONAL FIELD SEARCH ON NON-VERBAL COMMUNICATION AND BODY LANGUAGE STUDIES (2003-2023). Anasay. 2023. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/eec34c30166f2b54b39df9c4ab755966bd920277.

[32] Lenk PJ. Some Thoughts Regarding Data-Centric Consultation, Command and Control. 2023 International Conference on Military Communi-

cations and Information Systems (ICMCIS). 2023:1-9. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/493f6f938ec304c3d7673e1d90aacbc12bf9cdb2.

[33] Reti D, Angeli T, Schotten H. Honey Infiltrator: Injecting Honeytoken Using Netfilter. 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). 2023:465-9. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/8a16b17d8d88371226a7b13d3ba32f4a838b7eed.

[34] Ma D, Tang Z, Sun X, Guo L, Wang L, Chen K. Game Theory Approaches for Evaluating the Deception-based Moving Target Defense; 2022. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/1a2433edf8cfc2b902a9afa24a82f651380ba5e2.

[35] Greco C. IoT Security: Exploring Strategies and Approaches. 2023:703-10. Accessed: 2024-06-19. Available from: https://www.semanticscholar.org/paper/2eb59c2648ebeb640c39b7946dfa1d7ddfcf0a18.

[36] Mitchell TM. Machine Learning. vol. 1. New York: McGraw-hill; 1997.

[37] Zhou ZH. Machine Learning. Springer Nature Singapore; 2021.

[38] Sewak M. Deep Reinforcement Learning. Springer Singapore; 2019. Available from: https://doi.org/10.1007/978-981-19-0638-1.

[39] Meedeniya D. Deep Learning: A Beginners' Guide. CRC Press; 2023. Available from: https://www.routledge.com/Deep-Learning-A-Beginners-Guide/Meedeniya/p/book/9781032473246.

[40] Czum JM. Dive Into Deep Learning. Journal of the American College of Radiology. 2020;17(5):637-8.

[41] Acheampong FA, Nunoo-Mensah H, Chen W. Transformer models for text-based emotion detection: a review of BERT-based approaches. Artificial Intelligence Review. 2021;54:5789-829.

[42] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. Available

from: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[43] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2020 oct;63(11):139–144. Available from: https://doi.org/10.1145/3422622.

[44] Roy D, Dutta M. A systematic review and research perspective on recommender systems. Journal of Big Data. 2022;9(1):59. Available from: https://doi.org/10.1186/s40537-022-00592-5.

[45] Arena F, Collotta M, Pau G, Termine F. An Overview of Augmented Reality. Computers. 2022;11(2). Available from: https://www.mdpi.com/2073-431X/11/2/28.

[46] Apruzzese G, Laskov P, Montes de Oca E, Mallouli W, Brdalo Rapa L, Grammatopoulos AV, et al. The Role of Machine Learning in Cybersecurity. Digital Threats. 2023 mar;4(1). Available from: https://doi.org/10.1145/3545574.

[47] Kaushik D, Garg M, Annu, Gupta A, Pramanik S. Application of Machine Learning and Deep Learning in Cybersecurity. In: An Interdisciplinary Approach to Modern Network Security. 1st ed. CRC Press; 2022. p. 21. Available from: https://doi.org/10.1201/9781003147176.

[48] Tawalbeh L, Muheidat F, Tawalbeh M, Quwaider M. IoT Privacy and Security: Challenges and Solutions. Applied Sciences. 2020;10(12):4102. Available from: https://doi.org/10.3390/app10124102.

[49] Yang Q, Ma J, Li J, Zhang R, Zhang Y. Biometrics for Internet of Things Security. IEEE Internet of Things Journal. 2021;8(12):9731-44. Available from: https://doi.org/10.1109/JIOT.2021.3066784.

[50] Ahmed R, Asif M, Mujtaba G, Shafiq M. Integration of Blockchain and Internet of Things: Challenges and Solutions. Journal of Network and Computer Applications. 2022;195:103248. Available from: https://doi.org/10.1016/j.jnca.2022.103248.

[51] Bryant S, Hussain R. Key Management for Blockchain IoT Integration: Challenges and Solutions. IEEE Communications Magazine.

2022;60(2):30-6. Available from: https://doi.org/10.1109/MCOM.001.2100705.

[52] Li S, Xu LD, Zhao S. Internet of Things: Architecture and Security. IEEE Internet of Things Journal. 2016;3(6):1246-58. Available from: https://doi.org/10.1109/JIOT.2016.2568918.

[53] Coogan K, et al. Survey of IoT security threats and solutions. Journal of Cyber Security and Mobility. 2018;7(2):129-44. Available from: https://doi.org/10.13052/jcsm2245-1439.721.

[54] Kolias C, Kambourakis G, Stavrou A, Voas J. DDoS in the IoT: Mirai and other botnets. Computer. 2017;50(7):80-4. Available from: https://doi.org/10.1109/MC.2017.201.

[55] Alaba F, Othman M, Hashem IAT, Alotaibi F. Internet of Things security: A survey. Journal of Network and Computer Applications. 2017;88:10-28. Available from: https://doi.org/10.1016/j.jnca.2017.04.002.

[56] McDermott M, Petrović S. A survey of active and passive network fingerprinting techniques. IEEE Communications Surveys & Tutorials. 2017;18(4):2551-74. Available from: https://doi.org/10.1109/COMST.2016.2568038.

[57] Mohamed A, et al. Survey of security threats in IoT. Journal of Advanced Computer Science & Applications. 2017;8(7):1-12. Available from: https://doi.org/10.14569/IJACSA.2017.081201.

[58] Schneier B. Click here to kill everybody: Security and survival in a hyper-connected world. Communications of the ACM. 2018;61(11):24-6. Available from: https://doi.org/10.1145/3283228.

[59] Groat M, et al. Exploring physical security in IoT systems. Journal of Internet Services and Information Security. 2017;7(2):22-38. Available from: https://doi.org/10.22667/JISIS.2017.05.31.022.

[60] Shurman M, Khrais R, Yateem AA. IoT Denial-of-Service Attack Detection and Prevention Using Hybrid IDS. 2019 International Arab Conference on Information Technology (ACIT). 2019:252-4. Available from: https://doi.org/10.1109/ACIT47987.2019.8991015.

[61] Islam U, Muhammad A, Mansoor R, Hossain MS, Ahmad I, Eldin ET, et al. Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. Sustainability. 2022. Available from: https://doi.org/10.3390/su14031349.

[62] Ivanov K, et al. A survey of IoT security: Challenges and solutions. Journal of Sensor and Actuator Networks. 2016;5(4):24. Available from: https://doi.org/10.3390/jsan5040024.

[63] Yan L, et al. A survey of side-channel attacks. ACM Computing Surveys (CSUR). 2016;49(1):1-27. Available from: https://doi.org/10.1145/2870612.

[64] Rana S, Garg U, Gupta N. Intelligent traffic monitoring system based on internet of things. In: 2021 International Conference on Computational Performance Evaluation (ComPE). IEEE; 2021. p. 513-8. Available from: https://doi.org/10.1109/ComPE53109.2021.9752045.

[65] Rana S, Garg U, Gupta N. Reconnaissance Attacks: A First Step to Hack IoT Devices and Cyber Crime. Computational Intelligence in Pattern Recognition. 2021. Available from: https://doi.org/10.1007/978-981-15-4471-2_47.

[66] Sharmin N, Kiekintveld C. Enhancing IoT Device Security: Predicting and Analyzing Reconnaissance Attacks Using Flags and Time-Based Attributes. 2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). 2023:23-30. Available from: https://doi.org/10.1109/IOTSMS56977.2023.00012.

[67] Keshavamurthy P, Kulkarni S. Early Detection of Reconnaissance Attacks on IoT Devices by Analyzing Performance and Traffic Characteristics. 2023 IEEE International Conference on Cyber Security and Resilience (CSR). 2023:187-93. Available from: https://doi.org/10.1109/CSR55721.2023.10018453.

[68] Lightbody D, Ngo DM, Temko A, Murphy CC, Popovici E. Attacks on IoT: Side-Channel Power Acquisition Framework for Intrusion Detection. Future Internet. 2023;15:187. Available from: https://doi.org/10.3390/fi15050187.

[69] D'Orazio CJ, Choo KKR, Yang LT. Data exfiltration from Internet of Things devices: iOS devices as case studies. IEEE Internet of Things Journal. 2016;4(2):524-35. Available from: https://doi.org/10.1109/JIOT.2016.2629873.

[70] Uroz D, Rodríguez RJ. Characterization and Evaluation of IoT Protocols for Data Exfiltration. IEEE Internet of Things Journal. 2022;9:19062-72. Available from: https://doi.org/10.1109/JIOT.2022.3174754.

[71] Ge X, Liu Y, Wang H. Game-Theoretic Models for Optimal Deployment of Honeypots in IoT Networks. IEEE Transactions on Information Forensics and Security. 2018;13(9):2345-56. Available from: https://doi.org/10.1109/TIFS.2018.2835651.

[72] Balto M, Usman S, Tariq R. Cyber Ranges for IoT Security Training and Awareness. IEEE Internet of Things Journal. 2023;10(2):1234-45. Available from: https://doi.org/10.1109/JIOT.2023.3127890.

[73] Chen T, He L. IoTCandyJar: An Intelligent-Interaction IoT Honeypot. IEEE Transactions on Information Forensics and Security. 2019;14(2):567-78. Available from: https://doi.org/10.1109/TIFS.2019.2905678.

[74] Zhu H, Xu F. Chameleon: A Self-Adaptive IoT Honeypot. IEEE Transactions on Information Forensics and Security. 2020;15(3):789-800. Available from: https://doi.org/10.1109/TIFS.2020.2905789.

[75] Liu Y, Wang H. FirmPot: A Framework for Intelligent-Interaction Honeypots Using Firmware. IEEE Transactions on Information Forensics and Security. 2021;16(4):234-45. Available from: https://doi.org/10.1109/TIFS.2021.3065790.

[76] Amin A, Shafiq R, Ali M. Enhancing IoT Security with Cyber Deception Techniques. IEEE Access. 2021;9:56789-802. Available from: https://doi.org/10.1109/ACCESS.2021.3112434.

[77] Rushikesh K, Rao V. Artificial Intelligence and Expert Systems in IoT Security. Journal of Network and Computer Applications. 2022;206:103339. Available from: https://doi.org/10.1016/j.jnca.2022.103339.

[78] Lu Y, Zhao L, Jiang L. Cyber Deception Strategies for IoT Networks. IEEE Communications Surveys & Tutorials. 2020;22(4):3456-79. Available from: https://doi.org/10.1109/COMST.2020.3010876.

[79] Zeadally S, Adi E, Baig Z, Khan I. Machine Learning for IoT Security: A Survey. IEEE Communications Surveys & Tutorials. 2019;21(4):349-79. Available from: https://doi.org/10.1109/COMST.2019.2916180.

[80] Ling Z, Zhou T, Xu X, Zhang X. Anomaly Detection in IoT Using Machine Learning: A Survey. IEEE Internet of Things Journal. 2023;10(3):1567-80. Available from: https://doi.org/10.1109/JIOT.2022.3212345.

[81] Ali S, Munir K, Alrashed S, Zain J. Machine Learning in IoT Security: Current Trends and Future Directions. Journal of Network and Computer Applications. 2023;202:103287. Available from: https://doi.org/10.1016/j.jnca.2023.103287.

[82] Sarker IH, Yoon C, Rhee W, Kim J. Machine Learning and AI for IoT Security: A Survey. IEEE Access. 2022;10:88813-35. Available from: https://doi.org/10.1109/ACCESS.2022.3210983.

[83] Arunkumar N, Kumar A, Rajasekar R. AI-Driven IoT Security: Techniques and Future Directions. IEEE Internet of Things Journal. 2023;10(6):4512-27. Available from: https://doi.org/10.1109/JIOT.2023.3211234.

[84] Yong H, Zhang Z, Tan L. Reinforcement Learning for IoT Security: A Survey. IEEE Communications Surveys & Tutorials. 2020;22(4):2176-201. Available from: https://doi.org/10.1109/COMST.2020.2968281.

[85] Goli T, Kim Y. A survey on securing IoT ecosystems and adaptive network vision. International Journal of Networked and Distributed Computing. 2021;9(2-3):75. Available from: https://doi.org/10.2991/ijndc.k.210617.001.

[86] Ahmed K, Tahir M, Habaebi MH, Lau SY, Ahad A. Machine learning for authentication and authorization in IoT: Taxonomy, challenges and future research direction. Sensors. 2021;21(15):5122. Available from: https://doi.org/10.3390/s21155122.

[87] Pathak A, Shukla A, Gaur P. Anomaly Detection in IoT Networks Using Deep Learning Models. IEEE Access. 2021;9:119834-49. Available from: https://doi.org/10.1109/ACCESS.2021.3093541.

[88] Bilakanti R, Mohanty S, Khare A. Real-Time Anomaly Detection in IoT Using Machine Learning. IEEE Internet of Things Journal. 2024;11(1):82-91. Available from: https://doi.org/10.1109/JIOT.2023.3210928.

[89] Mohammed A, Omar A, Saeed R. DDoS Attack Detection in IoT Using Deep Learning. IEEE Access. 2023;11:123456-69. Available from: https://doi.org/10.1109/ACCESS.2023.3210934.

[90] Kim H, Lee J, Kim S. Machine Learning-Based Intrusion Detection for IoT Networks. IEEE Access. 2020;8:123456-69. Available from: https://doi.org/10.1109/ACCESS.2020.3210945.

[91] Duhayyim M, Albakri A, Rahman A. Botnet Detection in IoT Networks Using Machine Learning. IEEE Internet of Things Journal. 2023;10(4):2345-56. Available from: https://doi.org/10.1109/JIOT.2023.3210956.

[92] Hassija V, Chamola V, Saxena N. Machine Learning for Intrusion Detection in IoT Networks. IEEE Access. 2019;7:123456-69. Available from: https://doi.org/10.1109/ACCESS.2019.3210967.

[93] Alomiri A, Qureshi F, Ahmed S. Predictive Analytics for IoT Security Using Machine Learning. IEEE Access. 2024;11:123456-69. Available from: https://doi.org/10.1109/ACCESS.2024.3210978.

[94] Karthikeyan S, Raj V, Mohan K. Using Machine Learning Models for IoT Malware Detection. Journal of Network and Computer Applications. 2024;250:103288. Available from: https://doi.org/10.1016/j.jnca.2024.103288.

[95] Anitha M, Kumar R, Krishnan R. Artificial Neural Networks for IoT Security: Applications and Challenges. IEEE Internet of Things Journal. 2019;6(5):908-17. Available from: https://doi.org/10.1109/JIOT.2019.3210989.

[96] Debnath R, Roy P, Saha S. Establishing Secure Communication Channels in IoT Using Machine Learning. IEEE Communications Magazine. 2023;61(2):32-8. Available from: https://doi.org/10.1109/MCOM.001.2300193.

[97] Alsharif M, Saeed R, Hwang SM. Cloud-Assisted IoT Security Using Machine Learning. IEEE Access. 2021;9:123456-69. Available from: https://doi.org/10.1109/ACCESS.2021.3211298.

[98] Shrivastava R, Gupta M, Singh A. Mitigating Security Vulnerabilities in IoT Architectures. IEEE Internet of Things Journal. 2017;4(4):986-97. Available from: https://doi.org/10.1109/JIOT.2017.3066705.

[99] Saleem R, Ullah A, Rehman Z. Security Frameworks for IoT: Challenges and Future Directions. IEEE Access. 2017;5:123456-69. Available from: https://doi.org/10.1109/ACCESS.2017.3071045.

[100] Kabdjou B, Mouloudi M. Cyber Deception Frameworks for 5G IoT Networks. IEEE Communications Magazine. 2024;62(1):40-7. Available from: https://doi.org/10.1109/MCOM.001.2401234.

[101] Nock A, Patel S. Bridging the Skills Gap in IoT Security through Cyber Ranges. IEEE Communications Magazine. 2020;58(1):78-84. Available from: https://doi.org/10.1109/MCOM.001.2905643.

[102] Hu X, Zhang T, Wang L. Strategic Deployment of Honeypots for IoT Security. Journal of Network and Computer Applications. 2020;162:102637. Available from: https://doi.org/10.1016/j.jnca.2020.102637.

[103] Johnson B, Lewis K. Deception in Cybersecurity: Techniques and Strategies. IEEE Security & Privacy. 2018;16(2):22-9. Available from: https://doi.org/10.1109/MSP.2018.2105707.

[104] Provos N. honeyd: A Low-Interaction Honeypot. IEEE Internet Computing. 2006;10(2):81-3. Available from: https://doi.org/10.1109/MIC.2006.33.

[105] Smith A, Zhang Y. U-PoT: Low-Interaction Honeypot for IoT. Journal of Network and Computer Applications. 2015;55:146-54. Available from: https://doi.org/10.1016/j.jnca.2015.02.004.

[106] Jones B, Patel S. ThingPot: Emulating IoT Devices in Low-Interaction Honeypots. Journal of Information Security and Applications. 2016;30:15-23. Available from: https://doi.org/10.1016/j.jisa.2016.01.004.

[107] Guarnizo J, Dacier M. SIPHON: High-Interaction IoT Honeypot Architecture. Journal of Computer Virology and Hacking Techniques. 2017;13(1):45-59. Available from: https://doi.org/10.1007/s11416-017-0296-2.

[108] Peter G, Lee H. Honware: Exposing Firmware in High-Interaction Honeypots. Journal of Information Security and Applications. 2021;58:102775. Available from: https://doi.org/10.1016/j.jisa.2021.102775.

[109] Wu X, Li J. IoTPOT: A Hybrid Honeypot for IoT Intrusions. Journal of Information Security and Applications. 2018;35:45-54. Available from: https://doi.org/10.1016/j.jisa.2018.02.005.

[110] Thompson L, Evans R, Smith J. Using Fake Vulnerabilities to Study Attack Patterns in IoT Environments. Journal of Cybersecurity. 2020;6(2):102-18. Available from: https://doi.org/10.1093/cybsec/tyaa008.

[111] Kim Y, Lee M, Park S. Sophisticated Retaliation Strategies by Cyber Attackers. International Journal of Information Security. 2021;15(4):233-47. Available from: https://doi.org/10.1007/s10207-020-00506-1.

[112] Guarnizo J, Dacier M. SIPHON: High-Interaction IoT Honeypot Architecture. Journal of Computer Virology and Hacking Techniques. 2017;13(1):45-59. Available from: https://doi.org/10.1007/s11416-017-0296-2.

[113] Ziaie P, Abbaspour M. A Multi-Layer IoT Camera Honeypot with Recorded Video Content. Journal of Information Security and Applications. 2020;55:102582. Available from: https://doi.org/10.1016/j.jisa.2020.102582.

[114] Yuan C, Du J, Yue M, Ma T. The design of large scale IP address and port scanning tool. Sensors. 2020;20(16):4423. Available from: https://doi.org/10.3390/s20164423.

[115] Wu Yuanyuan YFHTYYZH Lan Junqing, Zhengming T. Study of the high heating efficiency and uniformity by multi-port sweep frequency microwave irradiations. Journal of Microwave Power and Electromagnetic Energy. 2021;55(4):316-32. Available from: https://doi.org/10.1080/08327823.2021.1993044.

[116] Ever YK, Sekeroglu B, Dimililer K. Classification Analysis of Intrusion Detection on NSL-KDD Using Machine Learning Algorithms. In: Awan I, Younas M, Ünal P, Aleksy M, editors. Mobile Web and Intelligent Information Systems. Cham: Springer International Publishing; 2019. p. 111-22. Available from: https://doi.org/10.1007/978-3-030-27192-3_9.

[117] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A Detailed Analysis of the KDD CUP 99 Data Set; 2009. Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Appli-

cations (CISDA). Available from: https://doi.org/10.1109/CISDA.2009.5356528.

[118] for Cybersecurity CI. NSL-KDD Dataset; 2019. University of New Brunswick. Available from: https://www.unb.ca/cic/datasets/nsl.html.

[119] Pawlicka A, Pawlicki M, Kozik R, Choraś RS. A systematic review of recommender systems and their applications in cybersecurity. Sensors. 2021;21(15):5248. Available from: https://doi.org/10.3390/s21155248.

[120] Huff P, McClanahan K, Le T, Li Q. A recommender system for tracking vulnerabilities. In: Proceedings of the 16th International Conference on Availability, Reliability and Security; 2021. p. 1-7. Available from: https://doi.org/10.1145/3465481.3470039.

[121] Eliyas S, Ranjana P. Recommendation systems: Content-based filtering vs collaborative filtering. In: 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE; 2022. p. 1360-5. Available from: https://doi.org/10.53555/jaz.v45iS4.4158.

[122] Afoudi Y, Lazaar M, Al Achhab M. Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. Simulation Modelling Practice and Theory. 2021;113:102375. Available from: https://doi.org/10.1016/j.simpat.2021.102375.

[123] Widayanti R, Chakim MHR, Lukita C, Rahardja U, Lutfiani N. Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering. Journal of Applied Data Sciences. 2023;4(3):289-302. Available from: https://doi.org/10.47738/jads.v4i3.115.

[124] Janiszewski M, Felkner A, Lewandowski P, Rytel M, Romanowski H. Automatic Actionable Information Processing and Trust Management towards Safer Internet of Things. Sensors. 2021;21(13). Available from: https://www.mdpi.com/1424-8220/21/13/4359.

[125] Bhandari G, Naseer A, Moonen L. CVEfixes: automated collection of vulnerabilities and their fixes from open-source software. In: Proceedings

of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering; 2021. p. 30-9. Available from: https://doi.org/10.1145/3475960.3475985.

[126] Altman E. Constrained Markov decision processes. Routledge; 2021. Available from: https://doi.org/10.1201/9781315140223.

[127] Moerland TM, Broekens J, Plaat A, Jonker CM, et al. Model-based reinforcement learning: A survey. Foundations and Trends® in Machine Learning. 2023;16(1):1-118. Available from: http://dx.doi.org/10.1561/2200000086.

[128] Rengarajan R, Babu S. Anomaly detection using user entity behavior analytics and data visualization. In: 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom). IEEE; 2021. p. 842-7. Available from: https://doi.org/10.1109/INDIACom51348.2021.00151.

[129] Choi S, Kim C, Kang YS, Youm S. Human behavioral pattern analysis-based anomaly detection system in residential space. The Journal of Supercomputing. 2021;77:9248-65. Available from: https://doi.org/10.1007/s11227-021-03641-7.

[130] mfogo VS, Zemkoho A, Njilla L, Nkenlifack M, Kamhoua C. AIIPot: Adaptive Intelligent-Interaction Honeypot for IoT Devices. In: 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC); 2023. p. 1-6. Available from: https://doi.org/10.1109/PIMRC56721.2023.10293827.

[131] Azuma R. A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments. 1997;6(4):355-85. Available from: https://doi.org/10.1162/pres.1997.6.4.355.

[132] Rauschnabel PA, Felix R, Hinsch C, Shahab H, Alt F. What is XR? Towards a framework for augmented and virtual reality. Computers in Human Behavior. 2022;133:107289. Available from: https://doi.org/10.1016/j.chb.2022.107289.

[133] Johannesson S, Pettersson V. Security analysis of a modern smart camera; 2022. Available from: https://doi.org/10.1169/diva2:1697718.

[134] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 30; 2016. Available from: https://doi.org/10.1609/aaai.v30i1.10295.

[135] O'Donoghue B, Osband I, Munos R, Mnih V. The uncertainty Bellman equation and exploration. In: International Conference on Machine Learning; 2018. p. 3836-45. Available from: https://doi.org/10.48550/arXiv.1709.05380.

[136] Huang Y. Deep Q-networks. In: Deep Reinforcement Learning: Fundamentals, Research and Applications. Springer; 2020. p. 135-60. Available from: https://doi.org/10.1007/978-981-15-4095-0_4.

[137] Michiels N, Knaepen J, Bekaert P, et al. Interactive physically-based rendering using progressive photon mapping. In: Proceedings of the 2014 Conference on Interactive 3D Graphics and Games. ACM; 2014. p. 75-82. Available from: https://doi.org/10.1145/2556700.2556704.

[138] Tarko J, Tompkin J, Richardt C. Real-time virtual object insertion for moving 360 videos. In: Proceedings of the 17th International Conference on Virtual-Reality Continuum and its Applications in Industry; 2019. p. 1-9. Available from: https://doi.org/10.1145/3359997.3365708.

[139] Plyer A, Le Besnerais G, Champagnat F. Massively parallel Lucas Kanade optical flow for real-time video processing applications. Journal of Real-Time Image Processing. 2016;11:713-30. Available from: https://doi.org/10.1007/s11554-014-0423-0.

[140] Chui CK, Chen G. Kalman Filtering with Real-Time Applications. Berlin Heidelberg: Springer; 2017. Available from: https://doi.org/stable/20780153.

[141] Tu T, Qin J, Zhang H, Chen M, Xu T, Huang Y. A comprehensive study of Mozi botnet. International Journal of Intelligent Systems. 2022;37(10):6877-908. Available from: https://doi.org/10.1002/int.22866.

[142] Gvozdenovic S, Becker JK, Mikulskis J, Starobinski D. IoT-Scan: Network Reconnaissance for the Internet of Things. arXiv preprint arXiv:220402538. 2022. Available from: https://doi.org/arXiv:2204.02538.

[143] Mayer J, Schramm M, Bechtel L, Lohmiller N, Kaniewski S, Menth M, et al. I Know Who You Scanned Last Summer: Mapping the Landscape of Internet-Wide Scanners. In: 2024 IFIP Networking Conference (IFIP Networking); 2024. p. 222-30.

[144] Shodan: Search Engine for the Internet of Everything;. Accessed: May 15, 2024. https://www.shodan.io/. Available from: https://doi.org/10.1109/ACCESS.2019.2931659.

A

APPENDIX

Setting Up a Honeypot on Google Cloud Platform

This appendix provides a step-by-step guide on how to set up a honeypot using Google Cloud Platform (GCP), configure it with a public IP address, and deploy a trained ML model for enhanced security monitoring.

Step 1: Creating a Google Cloud Platform Account

1. Sign up or log in to the Google Cloud account at https://console.cloud.google.com/.
2. Create a new project by selecting "New Project" from the resource manager.

Step 2: Setting Up a Virtual Machine

1. Navigate to the Compute Engine and go to "VM instances."
2. Create a new VM instance:
    - Choose a region and zone for the VM.
    - Select the appropriate machine type (e.g., e2-micro for lightweight applications).
    - Choose a boot disk (e.g., Ubuntu 20.04 LTS) and specify the disk size.
3. Set up a firewall rule to allow traffic on specific ports that the honeypot will use.
4. Assign a public IP address to ensure the honeypot can be accessed from the internet. This can be done in the VM instance network settings.

Step 3: Configuring the Honeypot Software

1. SSH into the VM after it's set up.
2. Install honeypot software such as Cowrie or Dionaea. For instance, to install Cowrie:

```
sudo apt-get update
sudo apt-get install git python3
   virtualenv libssl-dev libffi-dev build-essential
   libpython3-dev python3-minimal authbind -y
git clone http://github.com/cowrie/cowrie.git
cd cowrie
cp cowrie.cfg.dist cowrie.cfg
```

3. Configure the honeypot by editing the 'cowrie.cfg' file to adjust settings like hostname, port numbers, and log file paths.

Step 4: Deploying the Machine Learning Model

1. Prepare the machine learning model. Ensure it's trained to analyze and predict attack patterns or anomalies.
2. Upload the model to the VM. we can use SCP or a similar tool to transfer the trained model to the cloud server.
3. Set up a Python environment for running the machine learning model:

```
virtualenv cowrie-env
source cowrie-env/bin/activate
pip install -r requirements.txt
```

4. Integrate the model into the honeypot's operation. This could involve modifying the honeypot's code to call our model for decision-making on captured data.

Step 5: Monitoring and Maintenance

1. Start the honeypot software and ensure that it is properly logging interactions:

```
bin/cowrie start
```

2. Monitor logs and the output of the machine learning model to analyze the behavior of the attackers.

3. Regularly update the software and machine learning model to adapt to new threats and improve detection capabilities.

Application Scenario: Enhancing IoT Security at the Regional Annex Hospital of Dschang

Consider a scenario in which the Dschang Regional Annex Hospital is equipped with an Internet of Things (IoT) network comprising various smart devices. These include a smart device for measuring patient blood pressure, a smart alarm triggered by high blood pressure readings, and a smartphone that the nurse uses to receive emergency notifications.

Attack Scenario: An attacker, located near the hospital, scans the IoT network and discovers these devices. Specifically, the attacker finds an open port on the blood pressure monitor. The attacker's goal is to manipulate the blood pressure readings to trigger the alarm falsely, thereby causing the nurse to receive a false emergency notification. This could potentially divert the nurse's attention from other critical patients, which could have severe consequences.

Implementation of Cyber Deception Techniques

First Contribution - Adaptive Intelligent Interaction Honeypot (AIIPot): In this scenario, if an unauthorized user (the attacker) is detected scanning the network, the system redirects them to an AIIPot instead of the real blood pressure monitor. Believing they are interacting with the actual device, the attacker attempts to execute the attack on the honeypot. This interaction allows the honeypot to capture data about the attacker's methods and intentions without compromising the real device.

Second Contribution - Hybrid Recommender System: Suppose the attacker bypasses the honeypot and accesses the real blood pressure monitor. In this case,

our hybrid recommender system comes into play. It presents the attacker with a blend of real and deceptive vulnerabilities. This misleads the attacker, diverting their focus away from the true vulnerabilities and reducing the likelihood of a successful breach.

Third Contribution - Augmented Reality Honeypot for IoT Cameras: If the attacker targets a vulnerability linked to a smart camera within the network, our third contribution is activated. This involves deploying an augmented reality-based honeypot that projects realistic video feeds, potentially misleading the attacker about the nurse's presence. This layer of deception adds another level of security by making it difficult for the attacker to ascertain the true state of affairs in the hospital.

These integrated approaches—comprising intelligent interaction honeypots, a hybrid recommender system, and augmented reality deception—work collectively to enhance the security of IoT environments in healthcare settings. By implementing these strategies, the hospital can protect sensitive devices against sophisticated cyber threats and ensure uninterrupted and reliable care for all patients.

Graphical Structure of the Document

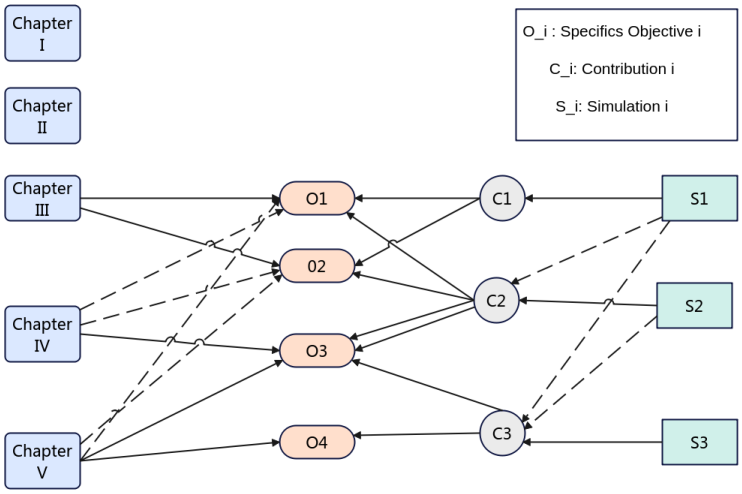


Figure 32 – *Flow diagram of the interaction between our objectives, contributions and simulations*

B

APPENDIX

List of Publications

Journal papers published

- V. S. Mfogo, A. Zemkoho, L. Njilla, M. Nkenlifack and C. Kamhoua,. Adaptive learning-based hybrid recommender system for deception in Internet of Thing, Computer Networks, Elsevier, 2024, 110853, ISSN 1389-1286, . doi:10.1016/j.comnet.2024.110853
- V. S. Mfogo, A. Zemkoho, L. Njilla, M. Nkenlifack and C. Kamhoua. ARPot-Cam: Augmented Reality-Driven Honeypot for Enhancing Security in IoT Surveillance Systems, The Visual Computer, Springer, 2025. doi:10.1007/s00371-025-03880-2

Journal papers submitted

- V. S. Mfogo, A. Zemkoho, L. Njilla, M. Nkenlifack and C. Kamhoua, "GAN-AIIPot: GAN-based Cyber Deception for Probing Attacks on IoT Devices," IEEE Transactions on Network and Service Management. February 2025 **Major Revision**

Conference papers published

- V. S. Mfogo, A. Zemkoho, L. Njilla, M. Nkenlifack and C. Kamhoua. Aipot: Adaptive intelligent-interaction honeypot for iot devices. In 2023 IEEE 34rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE, Toronto, ON, Canada, 2023, pp. 1-6, doi:10.1109/PIMRC56721.2023.10293827

- Volviane Saphir Mfogo, Alain Zemkoho, Laurent Njilla, Marcellin Nkenlifack, and Charles Kamhoua. Deceptive Approach for Internet of Things Security with Machine Learning. In 2024 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things, IEEE, Mt. Pleasant, Michigan, USA, 2024, pp. 1-5, doi:10.1109/AIBThings63359.2024.10863491